

The Influence of Neck Posture and Helmet Configuration on Neck Muscle Demands

Final Report
June 22, 2014

Prepared by:

Jack P. Callaghan PhD, CCPE
Professor and Canada Research Chair in Spine Biomechanics and Injury Prevention Department of
Kinesiology
Faculty of Applied Health Sciences
University of Waterloo
Waterloo, Ontario, Canada N2L-3G1
Voice: (519) 888-4567 x37080
Fax: (519) 746-6776
Email: jack.callaghan@uwaterloo.ca

Prepared for:

Ann Nakashima
Defence Scientist
Individual Behaviour and Performance Section Defence
Research and Development Canada Toronto Research
Centre
416-635-2000 ext 3064
ann.nakashima@drdc-rddc.gc.ca

PWGSC Contract number: W7714-125624/001/SV

The scientific or technical validity of this Contract Report is entirely the responsibility of the Contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Contract Report
DRDC-RDDC-2014-C188
April 2015

© Her Majesty the Queen as represented by the Minister of National Defence, 2014
© Sa Majesté la Reine, représentée par le ministre de la Défense nationale, 2014

TABLE OF CONTENTS:

EXECUTIVE SUMMARY	4
INTRODUCTION	6
<i>Participants</i>	<i>7</i>
<i>General Protocol</i>	<i>7</i>
Figure 1: Seating position for the MVE trials and simulated movement trials	7
Figure 2: Target head and neck postures	8
<i>Maximum Voluntary Exertions</i>	<i>8</i>
Table 1: Neck strength.....	9
Table 2: Reflective segment tracking marker locations.	10
Figure 3: Marker configuration for kinematic analyzes	10
<i>Task Simulation Tasks – Movement and Static Hold Trials</i>	<i>10</i>
Figure 4 Helmet, night vision goggle, and counterweight configurations	11
Figure 5: EMG electrode placement.....	12
<i>Data Processing and Dependent Variables</i>	<i>12</i>
Figure 6: Neck angle motion phase end points	15
Figure 7: Sample APDF curve	16
DISCUSSION OF RESULTS	17
<i>Neck Joint Strength</i>	<i>17</i>
Figure 8: Average neck strength in a neutral neck posture	17
Table 3: Average neck strength across posture conditions	18
<i>Neck Joint Motion (Kinematics).....</i>	<i>19</i>
Figure 9: Average three-dimensional peak neck angles.....	20
<i>Neck Muscular Response to Helmet Configuration</i>	<i>21</i>
Figure 10: Average peak EMG.....	23
Table 4: Percnt of APDF trials exceeding criterion values	23
Figure 11: APDF results.....	24
<i>Helmet Main Effects</i>	<i>25</i>

Figure 12: Average EMG magnitudes.....	26
Figure 13: APDF results.....	27
<i>Helmet by Posture Interactions</i>	28
<i>Neck Muscle Co-activation</i>	28
Figure 14: Peak EMG demonstrating statistical interactions.	29
Figure 15: Interaction plot of the fifth trial phase for MCI	30
Figure 16: MCI for the static hold phase of the movement trials.....	31
CONCLUSIONS	32
REFERENCES	34
APPENDICES	38
<i>Appendix A1:</i> Data file directory listing and MATLAB file structure descriptions	38
<i>Appendix A2:</i> MATLAB code for processing kinetic (Moment) data.	42
<i>Appendix A3:</i> MATLAB code for processing Kinematic (Motion) data.	64
<i>Appendix A4:</i> MATLAB code for processing EMG data.	92
<i>Appendix A5:</i> Full Statistical listing for EMG measures.	112

University of Waterloo Contributors:

Clark Dickerson PhD
Andrew Laing PhD
Caroline Joseph PhD
Colin McKinnon MSc
Mamiko Noguchi MSc

EXECUTIVE SUMMARY:

This report presents data collected from eight male participants and examines neck responses for postures designed to replicate movements associated with the flight crew of the CH146. Neck joint strength, neck joint motion (kinematics), and the muscular activation measured by Electromyography (EMG) of 10 neck muscles were assessed while performing 7 distinct movements wearing 6 different helmet configurations. With respect to posture, the main focus was muscular responses to the static holding phase of the 7 postures, evaluated over a 15 second period. Regarding helmet design, this study focused on how the activity of the neck muscles changed with the position of night vision goggles and the counter weight mechanism under different postural constraints of the head and neck. Additional deliverables included the analysis techniques and tools (i.e., MATLAB computer code) developed to analyze the generated data streams.

Overall, the consistent responses and lack of differences in neck kinematics during the movement trials confirm the rigorous control of postural conditions for all helmet configurations. Further, **helmet configuration had minimal influence on muscular demands during the low velocity movements and static holds evaluated.**

Neck strength differed primarily as a function of posture and exertion direction. In the neutral posture, axial rotation and lateral bend demonstrated lower joint strength than flexion or extension. The largest posture-related difference was for axial rotation conditions, as neck strength increased by 107% for a 45° axial rotation with 45° flexion posture compared to axial rotation in a neutral neck position. These findings imply that it would be beneficial to design work tasks to avoid these postures, especially in context of helmets **that increase load moments in non-neutral postures.** The lowest joint strength was in axial twist, indicating additional loading of the head and neck would have the greatest potential for negative consequences during tasks requiring moments to be created about this axis in isolation or in coupled movements.

Numerous data processing approaches and metrics were employed to assess the potential influence of helmet configuration on muscular responses of the neck. In general, the muscular demands were modest, with mean EMG values for all muscles and tasks ranging between 0.6 to 12.9% of Maximum Voluntary Exertion (MVE). Overall only 6 main effects of helmet were observed out of 315 comparisons. When these data were closely examined they demonstrated limited biological relevance with changes between the no helmet and helmeted conditions of less than 0.5 %MVE for mean and Root Mean Square (RMS) EMG. Amplitude Probability Distribution Function (APDF) magnitudes at the 10th, 50th, and 90th percentiles criterion values were all below standard occupational threshold limit values indicating minimal risk for muscle overload for the tasks and helmet conditions examined. There were many interactions between the helmet configurations and the postures adopted, but these were driven by differences between the no helmet control condition and the conditions with helmets. Importantly, when the no helmet condition was removed from the analyses, there were negligible differences between the different configurations within the helmeted conditions.

The six helmet conditions were evaluated in seven postural tasks. Movements were purposefully slow as per the directives of the original study to focus on the statically held postures. The

targeted postures created different muscular and joint demands associated with the 10 muscles evaluated, providing a useful range of postures to compare the demands associated with helmet configurations. The absence of kinematic differences across helmet conditions indicates that participants had similar task exposures when wearing each of the 6 helmet conditions, and if any differences were found in muscular responses they would be attributable to the helmet configuration and not variability in kinematics between task performances. Peak angles and total ranges of motion were similar across participants and helmet configurations, suggesting that the lack of helmet effects was not driven by differences in motion patterns across conditions.

While the muscular responses to the helmet conditions were quite low, it must be taken in the context of the condition studied. The movements in this study were of very low velocity with the instructed focus to be on the 15-second static hold phase. The addition of the masses to the helmets in these low acceleration movements had minimal impact on the response of the neck musculature. Future directions should focus on replicating the rate of movements, type of movements and number of movements that are present during in-flight activities. The current findings neither support the usage of a counter weight mechanism, nor do they discount its usage. The use of night vision goggles with or without the counter weight did not produce notable changes in muscular responses.

INTRODUCTION:

Neck pain and associated musculoskeletal injuries are substantial issues for occupations requiring the use of heavy helmets such as military pilots. Between 43 to 97% of military helicopter and aircraft pilots experience acute neck pain during or shortly after flight (Lange et al., 2011; Van Den Oord et al., 2010; Salmon et al., 2011). Indeed, the prevalence of neck injury in pilots ranges from 10 to 60% depending on the injury definition and the type of aircraft piloted (Aydog et al., 2004; Schall, 1989; Vanderbeek, 1988; Pippig and Kriebel, 2000; Petrén-Mallmin and Linder, 1999, 2001; Hämäläinen et al., 1993; Hendriksen and Holewijn, 1999). Specifically for Canadian Forces (CF) pilots and flight engineers, neck pain and muscle fatigue are common with over 80% of CH146 Griffon helicopter pilots and flight engineers reporting neck pain associated with in-flight task demands (Bridger et al., 2002). Chronic pain is also an issue, as De Loose et al. (2008) reported an 18.9% 1-year prevalence of self-reported neck pain in a survey of military aircraft pilots. Neck injuries create a major socioeconomic burden for injured personnel, their spouses, and society in general (Jennum et al. 2013). Consequently, there is a need to consider interventions to reduce the risk of work-related neck Musculoskeletal Disorders (MSD) for military pilots and flight engineers.

The constrained helicopter cockpit design requires awkward head and neck postures to scan terrain and to interact with the control display unit. Attachment of a night-vision goggle (NVG) system increases the total helmet mass and alters the centre of mass location. NVG are known to increase peak and cumulative neck loading, and demand extreme neck postures to compensate for a reduced field of view (Forde et al. 2011). NVG use increases the activation of dorsal neck musculature, raising potential for muscle fatigue, and thus risk of neck injury (Phillips et al., 1983). To balance the neck flexion moment created by the NVG and mitigate the elevated muscular demand, a counterweight (CW) is frequently attached to the posterior aspect of the helmet. In general, one theory for elevated risk of injury occurs when the occupational demand or exposure requires a higher percentage of an individual's musculoskeletal strength capacity.

The original primary purpose of this study was to provide neck muscular activation input data for a biomechanical model (Fraser, 2011). Specifically, this study measured the muscular activation of participants wearing several Canadian Forces (CF) helicopter helmet configurations in various neck postures. Postures and movements were prescribed in the original 2011 SoW and encompassed twenty-one (21) maximal voluntary exertions performed in seven (7) head and neck positions commonly assumed by CH146 pilots. These tasks were followed by a set of slow dynamic head movements between a neutral position and the same seven (7) positions while wearing six (6) helmet configurations. Outcome measures included kinematic (postural) analysis, surface electromyography (EMG), and external force measurement in the Maximum Voluntary Exertion (MVE) tasks. In the context of the current SoW (Nakashima, 2014), the data from the original SoW were analyzed with the purpose of delineating how the activity of the neck muscles changes as a function of helmet configuration (i.e. effects of NVG and counter weights) and the posture of the head and neck. As additional deliverables, the analysis techniques and tools (i.e., MATLAB computer code) generated in the analysis of EMG and kinematic data were supplied to DRDC and are described in the report appendices.

METHODS:

Participants

Eight males participated in the study (mean \pm standard deviation: age = 21.3 ± 1.7 years, height = 177.9 ± 6.8 cm, and body mass = 79.7 ± 11.5 kg). Participants were excluded based on pre-existing self-reported acute or chronic neck, shoulder, or upper back pain or injury, representing a health non-injured sample group. Each participant provided informed written consent and the study received ethics approval from the University of Waterloo's Ethics Review Committee. Participants were selected from a student population and did not have any prior experience with military helicopter equipment.

General Protocol

Participants were seated with the hips and chest restrained using straps to constrain all thorax movement other than the head and neck (Figure 1). The seating system consisted of a padded seat back that extended to approximately scapular level, and a seat pan that did not allow the participant's feet to touch the floor. The lateral edges of the seat pan were equipped with handles to allow further torso bracing. During experimental trials, participants wore a Canadian Forces CH146 Griffon helicopter helmet and various combinations of an anterior-mounted night-vision goggle (NVG) system and a 600 g posterior-mounted counter weight (CW). NVG could be set to a "down" (active) position as they would be when in active use in the field and an "up" (inactive) position as they would be when not in use. The study had two main components, an evaluation of the maximum voluntary exertion (MVE), or neck strength trials followed by a simulation of a sample of the postures typically held by CH146 flight personnel. Seven postures were tested in both phases of the study (Figure 2).



Figure 1: Seating position for the MVE trials (a) and simulated movement trials (b).

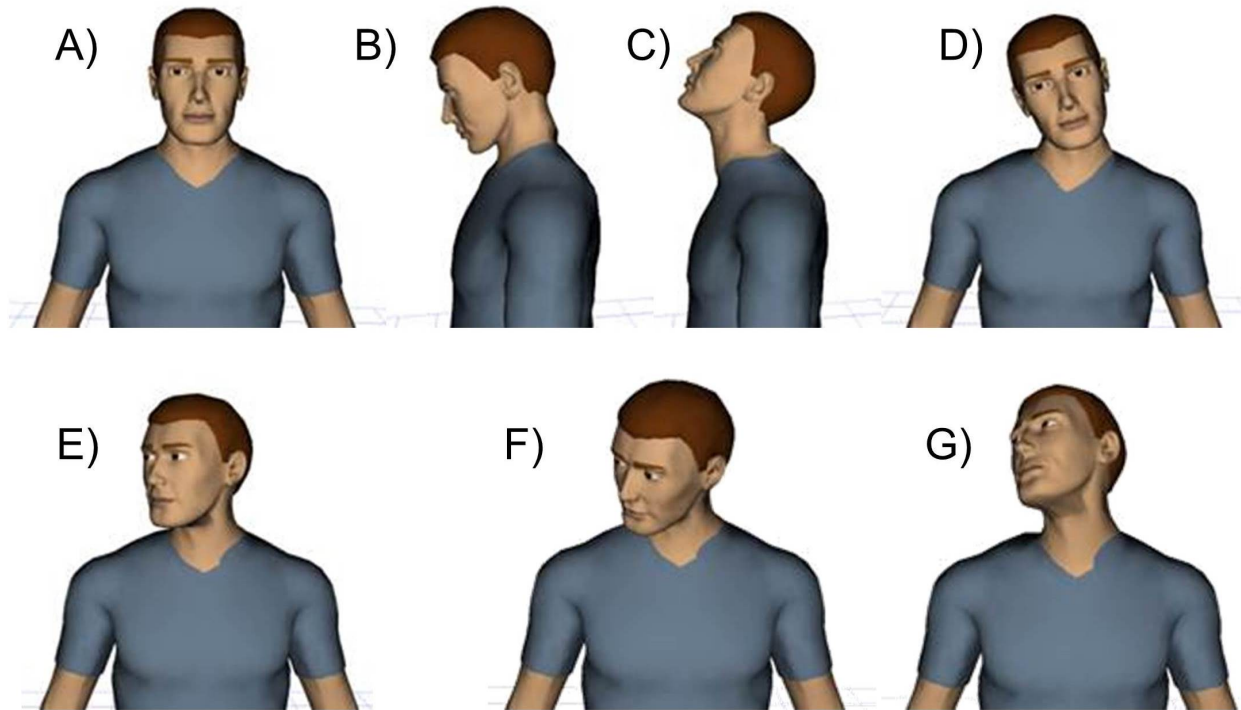


Figure 2: Target head and neck postures used for maximum voluntary exertion and experimental static hold trials: **A)** neutral (Neutral), **B)** 45 degrees of flexion (45Flex), **C)** 30 degrees of extension (30Ext), **D)** 20 degrees of rightward lateral flexion (20Lat), **E)** 45 degrees of rightward axial twist (45Rot), **F)** combination of 45 degrees of flexion and 45 degrees of rightward axial twist (45Rot/45Flex), and **G)** combination of 30 degrees of extension and 45 degrees of rightward axial twist (45Rot/30Ext).

Maximum Voluntary Exertions

Isometric maximal voluntary exertions (MVEs) were performed by participants in different postural conditions to determine direction-specific strengths. The session began with a 5-minute warm-up period that included passive neck and trunk stretching and submaximal static neck exertions. The participant was then positioned within the described seating system (Figure 1). MVEs were performed against a 6 degree-of-freedom load cell (MC3A-6-500, Advanced Mechanical Technology, Inc., USA) mounted between a helicopter helmet (military-style SPH-4B helicopter helmet, Gentex Corporation, USA) and a variable-position support on a serial robotic arm (HP50 with NX100 Controller; Yakasawa Motoman Robotics, 3530 Laird Road, Unit 3, Mississauga, ON). The outputs from the load cell were sampled at 1500 Hz. Three different mounts (for flexion/extension, lateral bending, and axial rotation) were used to attach the helmet to the load cell in accordance with the main axis of effort for the 10 different conditions (Table 1). Specifically, for the neutral posture conditions, participants maximally pushed their head forward (flexion), backward (extension), to the right (lateral flexion), and axially rotated their head to the right (clockwise). For the non-neutral postures (single axis conditions: 45° flexion, 30° extension, 20° lateral bending and 45° axial rotation; multi-axis conditions: 45° axial rotation with 45° axial flexion, 45° axial rotation with 30° extension), participants used a maximum effort to return to a neutral neck posture. All conditions were static.

Three trials were collected in each posture for a total of 30 trials. Each trial involved data collection for a duration of seven seconds including a 2-second ramp up period, a three second maximum sustained period, and a 2-second ramp down period. Two minutes of rest were provided between trials to minimize potential muscular fatigue effects.

Table 1: The neck strength directions tested within the seven postures.

Neck conditions	Direction of effort	Mount used
<i>Neutral neck conditions</i>		
Neutral Flexion	Flexion	Flexion/Extension
Neutral Extension	Extension	Flexion/Extension
Neutral Axial Rotation	Axial rotation toward left	Axial rotation
Neutral Lateral Bending	Lateral bending toward left	Lateral bending
<i>Single axis conditions</i>		
45° Flexion	Extension	Flexion/Extension
30° Extension	Flexion	Flexion/Extension
45° Axial Rotation	Axial rotation toward left	Axial rotation
20° Lateral Bending	Lateral bending toward left	Lateral bending
<i>Multi-axis conditions</i>		
45° Axial Rotation with 45° Flexion	Extension and axial rotation toward left	Axial rotation
45° Axial Rotation with 30° Extension	Extension and axial rotation toward left	Axial rotation

Motion Tracking Head and Neck Kinematics

A motion capture system was used to record upper thorax and head (helmet) kinematic data synchronously with surface electromyography. In order to calculate the maximal moments generated at the intervertebral joint between the seventh cervical vertebrae and the first thoracic vertebrae (C7-T1), the positions of the head and neck were measured using a motion capture system (Vicon MX, Vicon Motion Systems Ltd, Los Angeles, CA). For each participant, reflective markers were affixed to the helmet (top, each side over ear protection) (Figure 3a) and the torso (right and left acromion, C7, sternal notch, xiphoid process) (Figure 3b). These reflective markers on the participant and helmet (Table 2) were collected synchronously with an additional eight markers on the corners of the load cell. A standing T-pose was used as a reference posture for marker calibration. The three-dimensional coordinates of these markers were used to assess the relative position of the load cell compared to the helmet for the three mounting configurations. Raw marker coordinate data was used to calculate three-axis Euler angles between the upper thorax and head segments. An XYZ rotation sequence was used to follow International Society of Biomechanics (ISB) recommendations for intervertebral motion (Wu et al., 2002). The three-dimensional coordinates of the tracking markers (helmet and torso) were used to assess the position of C7-T1 intervertebral disc centroid (MacKinnon et al., 1993) compared to the load cell for each trial. Kinematic data were collected at 50 Hz and synchronized with the load cell output.

Table 2: Reflective segment tracking marker locations.

Number	Marker ID	Description
1	C7	C7 spinous process
2	SS	Suprasternal notch
3	XP	Xyphoid process
4	RAC	Right acromion process
5	LAC	Left acromion process
6	HR	Over the right ear cover of the helmet
7	HL	Over the left ear cover of the helmet
8	HT	Most superior point of the helmet along the sagittal midline
9	HX	Auxiliary point above the right ear cover of the helmet

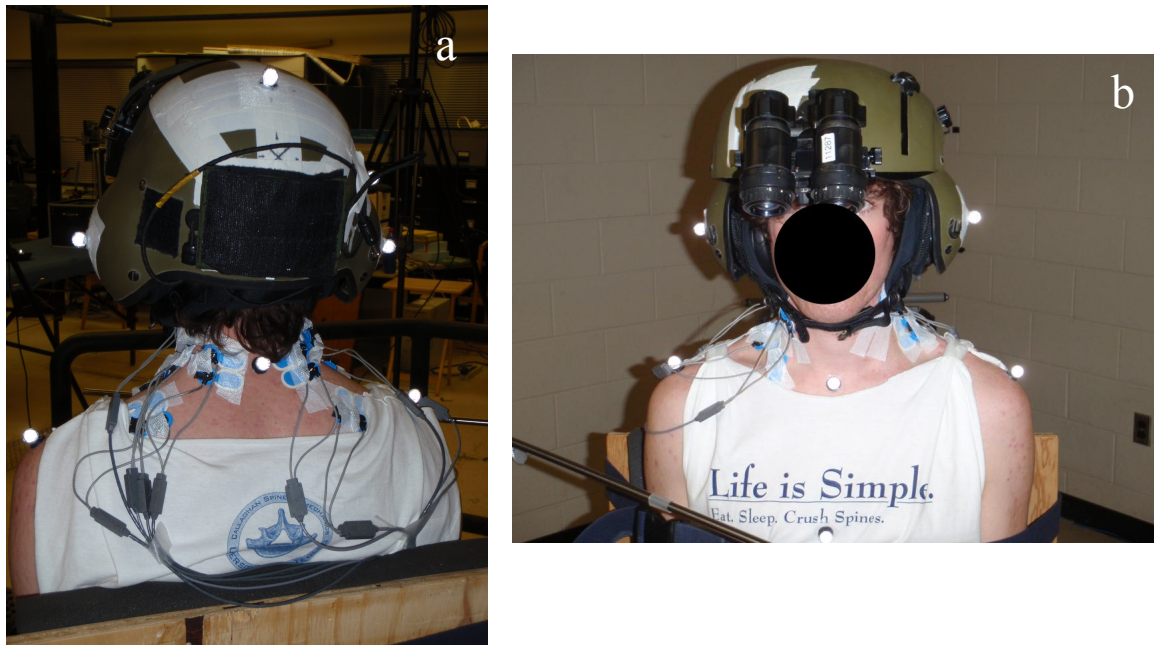


Figure 3: The reflective marker configuration for kinematic analyzes used for the helmet (a) and the participant torso (b).

Task Simulation Tasks – Movement and Static Hold Trials

Participants performed three trials in each combination of six helmet configurations and the same seven target head and neck postures in the MVE trials for a total of 126 experimental trials ($3 \times 6 \times 7 = 126$). For each trial, participants moved from a neutral posture to the marked target posture at a slow controlled pace allowing 2 s to complete the movements, held the end posture for 15 s, and then returned to neutral within a 2 s period. Target postures were confirmed in real-time with the kinematic motion tracking data as Euler rotation angles between the trunk and head

segments. Helmet configurations consisted of: 1) a control or no helmet condition; 2) the helmet only (hOnly); 3) the helmet with NVG system attached and set in both the up (hNVGup) and; 4) down (hNVGdown) positions; 5) and the helmet with CW attached and NVG attached in both the up (hCWup); 6) and down (hCWdown) positions (Figure 4). Target head and neck postures consisted of the same 7 neck postures as in the MVE tasks of flexion, extension, lateral flexion, axial twist, and combined postures (Figure 2).

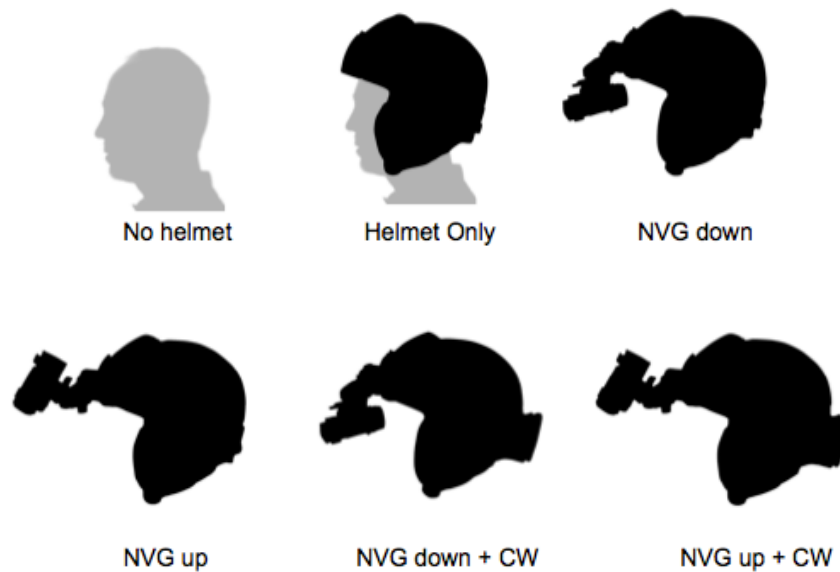


Figure 4: The experimental helmet, night vision goggle (NVG), and counterweight (CW) configurations.

Surface Electromyography (EMG)

Within the simulation tasks, muscular activation was evaluated throughout all trials. Fixed distance (2 cm), fixed pair EMG surface electrodes (Ag/AgCl; Noraxon USA, Inc., USA) were placed bilaterally on five muscles (for a total of 10 muscles) surrounding the neck: *m. splenius capitis* (SC), *m. sternocleidomastoid* (SCM), *m. levator scapulae* (LS), *m. cervical erector spinae* (ES), and *m. upper trapezius* (UT) (Figure 5). The skin was prepared using a disposable razor and an isopropyl alcohol wipe. The maximal voluntary exertions (MVEs) in each of the defined seven target head and neck postures were used to determine the maximal activations levels for EMG normalization. In each of the six non-neutral postures, participants attempted to return to a neutral neck posture against mechanical resistance using maximal effort. For the neutral posture, participants performed separate trials attempting to flex, extend, lateral bend to the right, and axial twist to the right using maximal effort, resulting in 10 total MVE trials per set. Each ramped MVE was recorded for 7 seconds and the peak EMG amplitude for each muscle across all MVE trials was used to normalize the static hold trial data. Signals were collected at 1500 Hz and amplified using a 16-channel Noraxon Telemetry 2400T G2 Telemetry electromyography system (Noraxon U.S.A. Inc., Scottsdale, AZ) with a common-mode rejection ratio of >100 dB.

Surface EMG provides a method that allows a larger detection volume around the electrode sites giving a representation of the activity of a muscle required during a task. When muscles are deep or further from the skin surface, such as beneath other muscles, surface EMG is not able to distinguish that muscle signal from the more superficial muscle. Surface EMG can also be prone to cross-contamination from adjacent muscles given the larger detection volume. In contrast indwelling EMG (fine wire electrodes) can be very specific in a localized area and are able to measure muscle activity in deep muscles. However, the detection volume is quite small and can be hampered by not sampling from a representative sample of motor units, thereby only giving a regional response of a muscle and those motor units near the electrode insertion site.

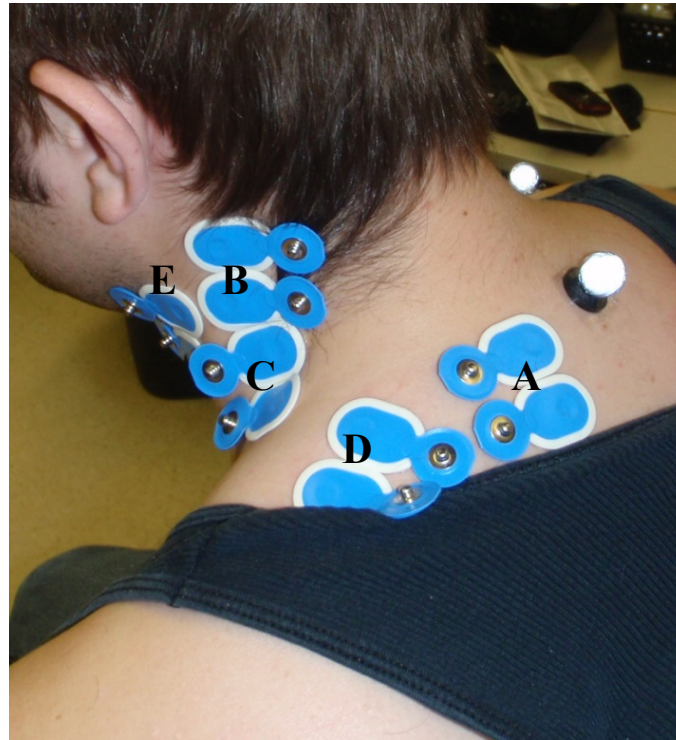


Figure 5: The location of electrode placement on a participant's left side for the five bilateral muscles recorded: **A)** Erector spinae (ES); **B)** Splenius capitis (SC); **C)** Levator scapulae (LS); **D)** Upper trapezius (UT); **E)** Sternocleidomastoid (SCM).

Data Processing and Dependent Variables

Following data collection, the kinetic (moment), kinematic (motion) and EMG data were post processed to evaluate the influence of postures and helmet configurations.

Kinetic – Neck Joint Moments

Load cell and kinematic data were smoothed using a zero-lag fourth order digital Butterworth low-pass filter with a cut-off frequency of 3 Hz. The marker data were up-sampled using linear interpolation to match the sampling frequency of the load cell output. Calibrations to the load cell outputs were made to account for the gravitational effects of the helmet and mounting hardware. The corrected force outputs were then rotated from the load cell coordinate system to the neck coordinate system. The torso coordinate system used was based on the International

Society of Biomechanics recommended thorax coordinate system (Wu et al. 2005) (x-axis: forward; y-axis: upward; z-axis: laterally toward the right) but with the origin at the C7-T1 joint instead of the sternal notch. Following this, the moments about C7-T1 were calculated based on the force outputs and distances between the origin of the load cell and C7-T1. Only the moment around the principal axis of effort for each posture condition was calculated and, for the complex posture conditions, the resultant moments of the two principal axis of effort were calculated to represent the effort direction that the participants were instructed to generate within the rigidly fixed helmet.

Two separate analysis approaches were used to characterize neck strength as parameterized by moments: i) the maximum of a 1-second averaged moving window, and ii) the instantaneous trial maximum. For each posture condition, the maximal value across the three trials was used for statistical comparisons.

Kinematic - Motion Analysis

The movement simulation trials were separated into five phases of movement (Figure 6): 1) pre-initiation, 2) initiation, 3) static hold, 4) return, and 5) post-return. To separate these movement phases, movement was defined as when the angular velocity in the primary direction of movement increased above 3 °/s. The pre-movement phase (1) was defined as the start of the trial to the point when movement began (angular velocity increased above 3 °/s). The initiation phase (2) was from when movement began until the point when movement stopped (angular velocity decreased below 3 °/s). Similarly, the return phase (4) was defined as start of movement to end of movement when returning to a neutral posture, and the post-return phase (5) was after movement stopped in a neutral posture to the end of the trial. The static phase (3) was the period between the initiation and return phases. The kinematic data were reduced by taking the peak values from each of the five phases within each trial as well as the range of motion (RoM). Since posture was a controlled variable (independent), the analyses of the kinematics evaluated whether similar movement patterns existed across helmet conditions. These data could be paired with the EMG data to perform analyses of more detailed neck joint kinetics once a suitable biomechanical joint model is identified or created.

Surface Electromyography (EMG)

All raw surface EMG signals were filtered to reduce any ECG contamination with a 30 Hz dual pass 2nd order high pass filter (Drake & Callaghan, 2006). These data were then linear enveloped (full-wave rectified and low pass filtered with a single-pass, 2nd order Butterworth filter) with a cut-off frequency of 4 Hz (Lu & Bishop, 1996). Smoothed data were normalized to maximum activity across all MVE trials. Except where noted, the following variables and processing approaches were evaluated for *each of the five trial phases* identified:

- *Time to maximum*
Time to maximum activity was calculated as the absolute time from the beginning of a movement phase to the maximum muscle activity during that phase.
- *Time to onset of activation*
Time to onset set of activity was calculated prior to the static hold (3), return (4), and post-return (5) phases of movement. For each of the ten recorded muscles (5 from each

side of the body), mean and standard deviation of muscle activity during a window 500 ms to 250 ms prior to the beginning of the phase were calculated. Onset of activity was identified when activity exceeded this baseline mean plus 2 standard deviations (Rietdyk et al., 1999) during a window 250 ms to 0 ms prior to the beginning of the phase based on published onset times for neck muscles being up to 175 ms (Magnusson et al., 1999; Brault et al., 2000; Sacher et al., 2012). Time to onset of activity was then defined as the amount of time between this onset of activity and the beginning of the phase. Time to onset of activity was not calculated prior to the initiation phase as the pre-initiation phase was normally less than 500 ms and onset time calculations could not be performed.

- *Maximum amplitude (peak)*
The maximum activation amplitude for each muscle.
- *Average amplitude (mean)*
The average activation amplitude for each muscle.
- *Median amplitude*
The median activation amplitude for each muscle.
- *Root-mean-square (RMS)*
The RMS activation amplitude for each muscle.
- *Muscular Co-activation Index (MCI) of the ten muscles*
In order to evaluate the overall net activation and involvement of the musculature surrounding the neck, an index to evaluate the weighted co-activation of the ten monitored muscles was employed. The value was based on the calculation of a net Euclidean magnitude of the RMS muscle activation with the following equation, where N is the 10 muscles, providing a greater influence of muscles that had higher activation in response to the task.

$$\text{Euclidean magnitude} = \sqrt{\sum_{i=1}^N \text{RMS activation}_i^2}$$

Amplitude Probability Distribution Function (APDF)

In addition to the seven EMG variables listed above, Amplitude Probability Distribution Functions (APDF) (Jonsson, 1978) were assembled from the EMG normalized linear envelope data across the entire trial length (i.e. without breaking out of the five trial phases) to yield muscular demands and risk during the simulation trials (Figure 7). The 10th, 50th, and 90th percentiles were calculated (APDF).

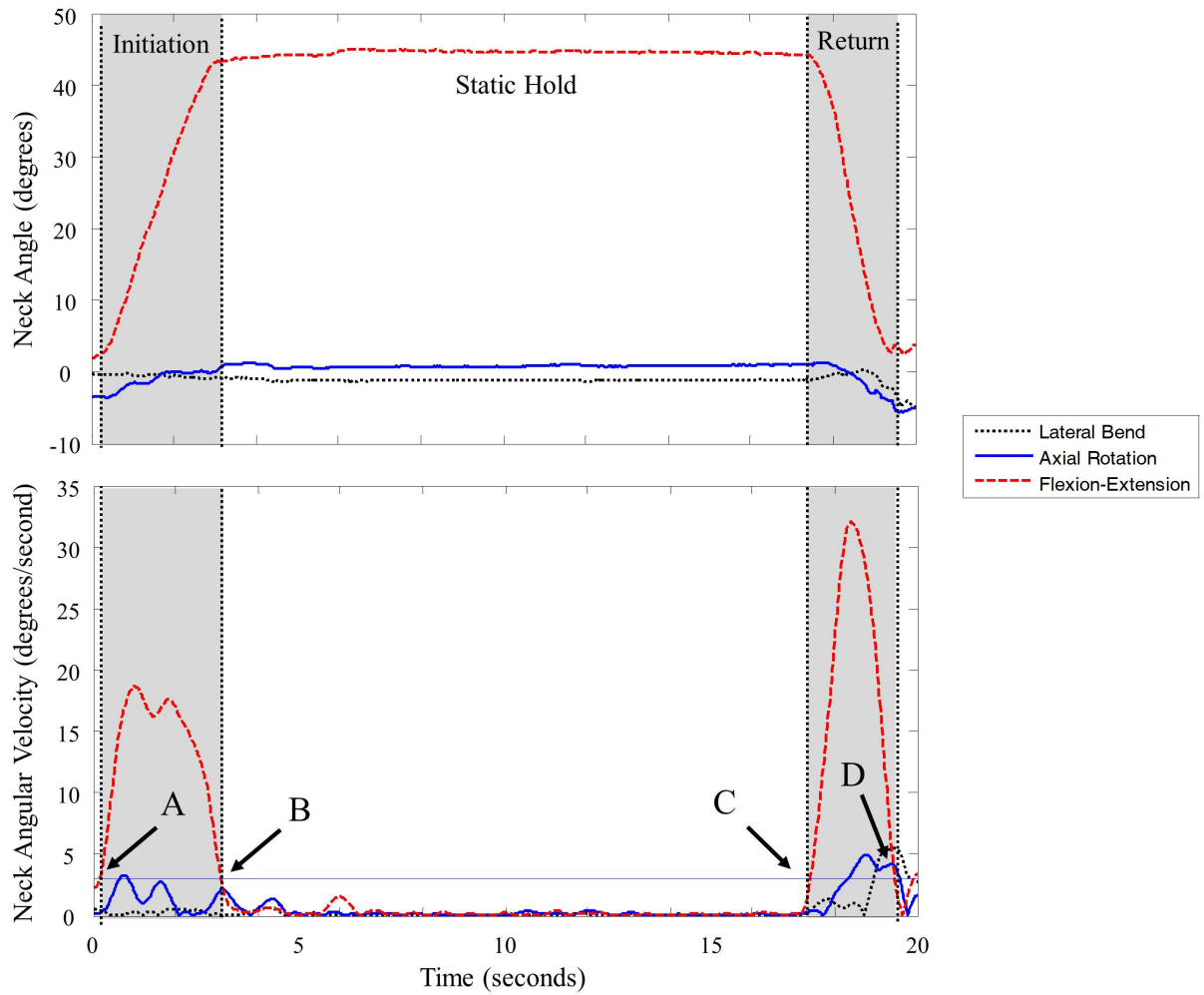


Figure 6: Sample 45° flexion trial neck angle and angular velocity data used to define motion phase end points. The lower pane shows where angular velocity crosses the movement threshold to determine: **A)** the end of the pre-movement phase; **B)** the end of the initiation phase; **C)** start of the return phase and **D)** end of the return phase. The 5 trial phases were defined as: **1)** from start of trial to A; **2)** from A to B; **3)** between B and C; **4)** from C to D; and **5)** from D to end of the trial. The upper pane shows these end points mapped onto neck angle data.

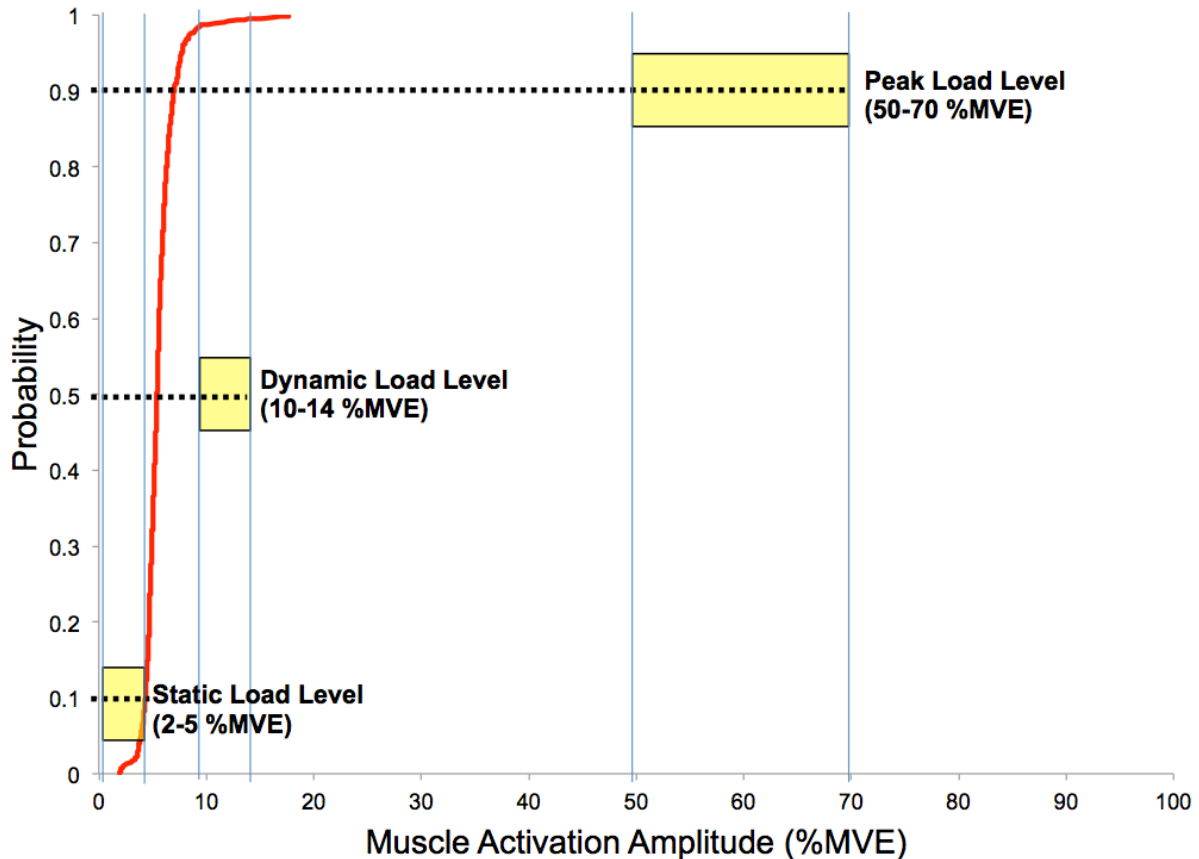


Figure 7: A sample APDF curve for the left cervical erector spinae (ES) in a helmeted trial with night vision goggles in the up position and no counterweight (hNVGup) for one participant. The criterion values from Jonsson (1978) are indicated as the center dotted line in the yellow boxes. If a muscle's amplitude crosses the threshold value to the right of the box at the dotted line it would be classified as exceeding the limit. For the plotted example, the ES is at the upper margin, but not exceeding 5%MVE for the 0.1 or 10% criterion. The muscle is well below the other two criterion values, falling to the left of the limits indicating lower activation levels.

Statistical Analyses

Two-factor repeated measures analyses of variance (ANOVAs) were used to assess the potential influence of posture or exertion direction (two to four levels) and analysis approach (maximum moving window vs. instantaneous maximum) on neck strength. Separate ANOVAs were performed for: i) flexion strength (two postures); ii) extension strength (two postures); lateral flexion (two postures); iv) axial rotation (four postures); and v) neutral postures (four directions). For the neutral posture and axial rotation analyses, Huynh-Feldt degree of freedom correction factors were used if sphericity assumption were not met. If main effects of posture were observed, post hoc analyses were completed using least square difference (LSD). For the EMG and kinematic data two-way repeated measures analyses of variance (helmet, posture) were conducted with LSMeans post-hoc test and Tukey corrections employed. All data were analyzed using SAS 9.3 (SAS, Cary, NC). Statistical significance was set at $\alpha = 0.05$.

DISCUSSION OF RESULTS:

Neck Joint Strength

To evaluate the neck joint strength the neck joint moments of 10 different exertion directions were examined. The direction of effort while in a neutral posture, the posture in which the MVE was performed, and the analysis approach used to characterize neck strength all influenced the magnitude of the maximal neck joint moment. There were no significant interactions between factors. Neck strength differences between 10 to 19 Nm depended on exertion direction when the neck was in a neutral posture ($p = 0.002$) (Figure 3). Post-hoc tests indicated that axial rotation strength was lower than extension strength ($p = 0.004$), flexion strength ($p = 0.013$) and lateral bending strength ($p = 0.017$) (Figure 8). However, neck strength in the extension, flexion and lateral bending directions were not different from each other.

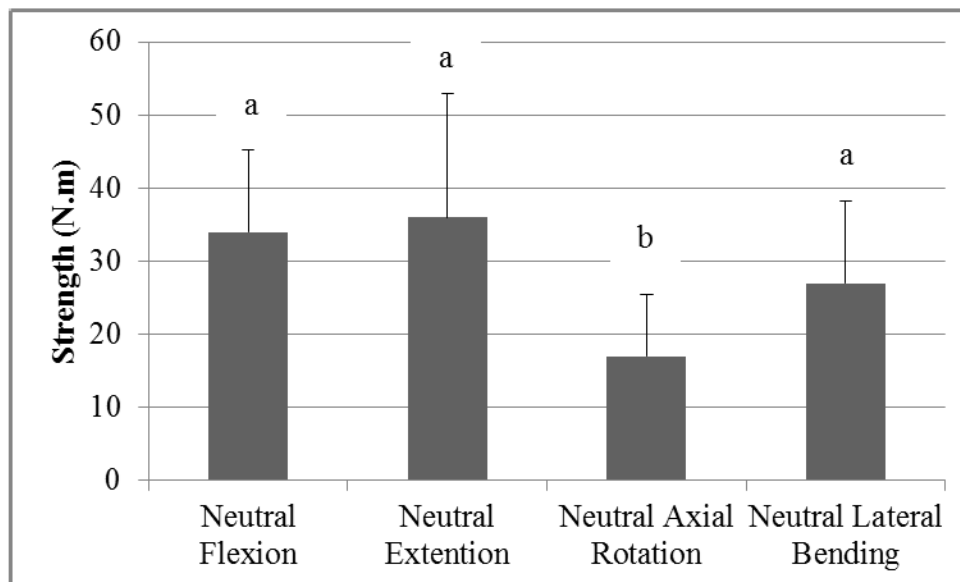


Figure 8: Average (+1SD) neck strength for four directions of effort while in a neutral neck posture. Conditions marked with different letters are significantly different ($p < 0.05$).

For non-neutral postures, differences in neck strength existed only for postures incorporating axial rotation exertions with differences ranging from 28 to 40 Nm ($p < 0.001$) (Table 3). Specifically, post-hoc analysis demonstrated that neck strength in the 45° axial rotation with 45° of flexion posture was larger than the other axial rotation postures conditions (68-107% difference). Posture did not influence neck strength for the flexion ($p = 0.35$), extension ($p = 0.59$) or lateral bending ($p = 0.20$) exertions.

For all postures, as would be expected, the strength values derived using the maximum instantaneous moment approach had significantly greater magnitude compared to the moving window approach ($p < 0.002$). Absolute differences ranged between 1 and 3 Nm with associated percentage differences ranging between 3 and 12 % (Table 3).

Table 3: Average and one Standard Deviation (the number in brackets (SD)) of neck strength across posture conditions. * indicates significant effects of posture and # indicates significant effects of strength analysis approach ($p < 0.05$). For the axial rotation exertions, posture conditions with the different letters are significantly different.

Posture conditions	Maximal 1-second moving window moment (Nm)	Maximum instantaneous moment (Nm)
<i>Flexion</i> [#]		
Neutral Flexion	34 (11)	35 (11)
30° Extension	31 (10)	32 (10)
<i>Extension</i> [#]		
Neutral Extension	35 (18)	37 (18)
45° Flexion	38 (15)	39 (14)
<i>Axial rotation</i> * [#]		
Neutral Axial Rotation ^a	16 (9)	18 (10)
45° Axial Rotation ^a	26 (21)	28 (22)
45° Axial Rotation with 45° Flexion ^b	55 (21)	58 (21)
45° Axial Rotation with 30° Extension ^a	27 (18)	28 (18)
<i>Lateral bending</i> [#]		
Neutral Lateral Bending	26 (10)	27 (10)
20° Lateral Bending	24 (10)	25 (11)

Neck strength was influenced by: i) the direction of the effort while in a neutral neck posture and that ii) neck postures significantly altered the strength capability when attempting to return to a neutral posture.

Neck strength, when the neck is in a neutral position, depends on the effort direction, with exertions in the sagittal plane being largest. Specifically, while there was no significant difference between extension and flexion strength in the neutral posture, these values were 72% (19 Nm) and 67% (17 Nm) larger than axial rotation, respectively. In addition, the values in lateral bending were 45% (10 Nm) smaller than the values in neutral axial rotation.

There were significant neck strength differences between neutral and complex postures. However, this was limited to axial rotation strength, with the combination of axial rotation and flexion (i.e. 45° axial rotation with 45° flexion) being significantly larger than the other conditions, i.e. 107% (39 Nm) greater than neutral axial rotation, 71% (30 Nm) greater than 45° axial rotation and 68% (29 Nm) greater than 45° axial rotation with 30° extension. These results were not unexpected as the more complex axial rotation conditions also involved effort in flexion or extension, which have higher strength capacity resulting in a coupled effort that represents the trends in the dominant flexion/extension axis that had an increase in strength at 45° flexion.

When the results from this study are compared with published studies employing a similar joint centroid (C7 or C7-T1), the results are within the range of maximal strength in neutral flexion (i.e. 12-53 Nm (Berg et al., 1994; Jordan et al., 1999; Mayoux-Benhamou and Revel, 1993; Seng

et al., 2002; Vasavada et al., 2001)), in neutral extension (i.e. 22-60 Nm (Staudte and Dühr, 1994; Queissere et al., 1994; Berg et al., 1994; Harms-Ringdahl et al., 1989; Jordan et al., 1999; Seng et al., 2002; Vasavada et al., 2001)), and in neutral lateral bending (i.e. 26-27 Nm (Vasavada et al., 2001; Seng et al., 2002)). However, the current study identified slightly higher axial rotation strengths than previous reports (8-11 Nm (Berg et al., 1994; Moroney et al., 1988; Strimpakos et al., 2004; Vasavada et al., 2001; Ylinen et al., 1999, 2004).

Neck Joint Motion (Kinematics)

The six helmet conditions were evaluated in seven postural tasks that were provided in the original SoW as being representative of the movements and postures occurring in the CH146 flight crew. Movements were purposefully slow as per the directives to focus on the static hold postures dictated. The rate of movement was a maximum of approximately 20 °/s in the tasks having end postures of 45°, with movement and return time set at 2 s to allow the participant to reach the targeted position. The static end posture was held for 15 s followed by another 2 s to allow for the return to the neutral starting position. The participants were given a target to guide the acquiring of the end point and neck joint positions were monitored in real time using the kinematic data. Since posture was a controlled variable, there were no expected differences to be present between the helmet conditions. However, the conditions did create statistically different end postures ($p < 0.05$) indicating that the postures assumed were different from each other, as expected given the range of postures selected to cover the primary and coupled movements of the neck. The statistical analyses supported this hypothesis with none of the helmet conditions being different ($p > 0.05$) within any of the 5 trial phases. This indicates that the participants had similar task exposures for each of the 6 helmet conditions, and any differences in the following EMG results appear to be attributable to helmet configuration changes and not any systematic differences in movements. The peak postures (Figure 9), neutral starting posture and the RoM were the same across participants and helmets. Further, the helmets did not introduce any “off-axis” differences. The three-dimensional angles were examined about all three rotational axes and there were no statistically significant responses in the axes that were secondary or tertiary to the targeted movement. For example if the movement trial was targeting 45° of flexion, the axial twist and lateral bend angles were also assessed and no differences were found. Further there were no interactions found between helmet and posture, indicating that the helmet did not introduce any differences in the way participants moved between the conditions.

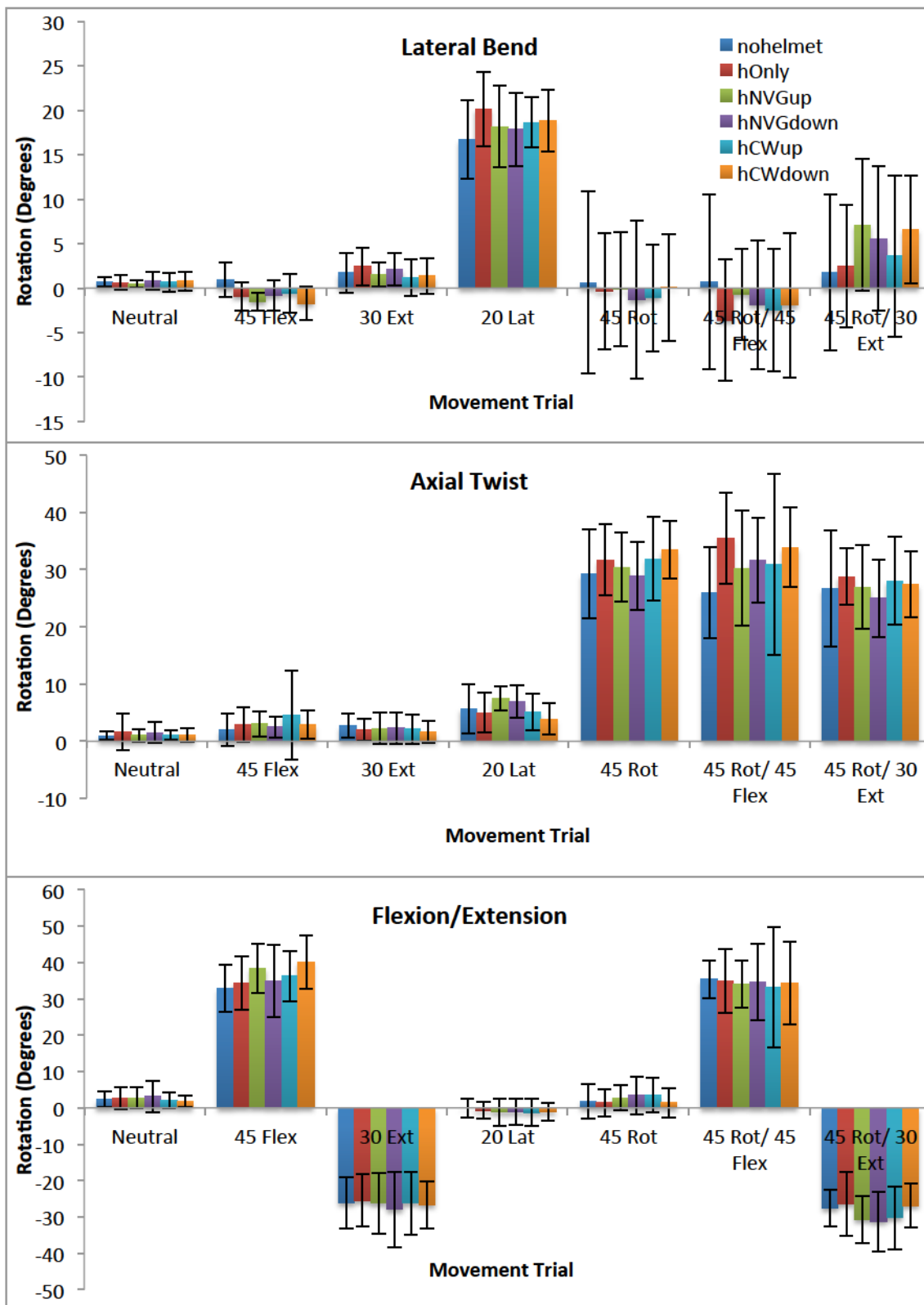


Figure 9: The average three-dimensional peak neck angles (\pm 1 standard deviation) achieved across participants in the static hold phase (phase 3). The three angles demonstrate similarity across the primary and resulting off-axis movements for the 6 helmet conditions.

Neck Muscular Response to Helmet Configuration

Summary: A large number of data processing approaches were employed and variables to assess any impact on the muscular responses of the neck to the helmet configurations examined. Overall 315 statistical effects were examined. This includes 250 from Mean, Median, Peak, RMS, Time to Maximum (5 variables by 5 trial phases by 10 muscles), 30 from the APDF analyses (3 variables by 10 muscles) and 5 from the weighted co-activation measure (5 trial phases). Of the 315 statistical comparisons only 6 main effects of helmet were found where no interaction effects were present. In contrast, there were 83 posture by helmet interactions (Appendix A5 includes a complete listing). As expected the main effect of posture produced frequent significant differences, but this was an independent variable designed to elicit different postures so this effect was not examined unless contributing to an interaction (i.e. indicating that the helmet configuration introduced a differing response across changing postural demands). Overall the muscular demands were modest for all muscles and tasks (Figure 10). The length of exposure during the static holds was 15 seconds and the mean and peak EMG levels would be considered below relevant occupational ergonomics standards. During the static hold phase of the movement trials, mean muscle activation levels ranged from 0.6 to 12.9% MVE across all conditions. This is comparable to EMG mean for neck extensor musculature reported during computer mouse and keyboard work of 2-7%MVE (Laursen et al., 2002) and 2-4%MVE during urban bus drivers (Leinonen et al., 2005). The highest peak muscle activation recorded was 21.7% MVE for any single muscle in any condition (SCM-L). More detailed information is provided in the following sections.

Across muscle comparisons revealed that the sternocleidomastoid (SCM) had the highest muscular activations. When the average magnitudes of the APDF were examined for the 10th, 50th, and 90th percentiles criterion values there were no values that exceeded the suggested limits (Jonsson, 1978) of 2-5% MVE for the 10th percentile, 10-14% MVE for the 50th percentile and 50-70% MVE for the 90th percentile (Figure 11) indicating minimal risk for muscle overload for the tasks and helmet conditions as employed in this study.

These findings must be considered both within the context of the study design and within the APDF data processing approach. First, the study design only assessed slow movements, allowing 2 seconds for the participants to achieve the end positions followed by static holds of the positions for 15-second periods. These tasks were performed for the 6 helmet conditions and 7 postures, with each combination repeated 3 times for a total of 126 task trials per participant. This accounts for a total “working” exposure of 42 minutes, with static holds accounting for 32 minutes of exposure. The study was a single exposure to the helmet conditions and not a chronic usage or longer exposure period that flight crews would experience. This study examined the immediate response of healthy pain-free participants to the changing conditions of the helmet configuration. Further, participants were given rest between each task trial; further offsetting any potential fatigue related issues that might arise in field usage. The experiments were performed in a quiet isolated lab, with no secondary task exposures that have been linked to elevated muscular activity responses. Both dual task demands and situational stress have been shown to result in increased muscular activation. Mental workload increased activation and decreased muscular rest in the trapezius muscle during computer data entry tasks (Schleifer et al., 2008) and time constrained stress situations increased static muscle activity of the neck extensors during

computer work (Laursen et al., 2002). The helmet task trials were all conducted in a quiet room, which does not replicate the whole body vibration that the flight crew would experience while wearing the helmets. Fundamental animal studies have demonstrated that whole body vibration elevated the neck muscular response in rabbits (Minasian et al., 1989). Similarly, a localized vibration applied to the trapezius muscle elicited an acute response of increased activation magnitude (Aström et al., 2007), although the chronic impact of this exposure was not evaluated.

The APDF processing approach was developed to assess prolonged occupational exposures in order to provide a mapping of the distribution of muscular efforts required over a work shift. The trial length employed in this study of 20 seconds does not directly map well onto this method's foundation. However, when the trials and participant data are collapsed, it provides a larger data sample to assess the responses across participants and conditions. When the individual trials for each participant were evaluated (Table 4), there were a small number of trials that exceeded the criterion values. Again, this must be taken in the context that a contraction violating the 10th percentile criterion simply means the muscle had contractions within the 20 second trial over 5%MVE for 2 seconds. This does not translate directly into a risk due to the short exposure, and more consideration should be placed on the mean findings across participants given the short trial length. There were no peak activations exceeding the 90th criterion and only two muscles at the 50th level. Given the novel task exposure, the short trial length and the large number of total trials (1008), this small percentage of trials exceeding the thresholds further supports the low muscular response to the helmeted condition in the conditions tested. What these findings highlight is that the sternocleidomastoid and upper trapezius were the two muscles most involved in the tasks and should be a focus for future evaluations.

The ideal evaluation would encompass prolonged exposures replicating at minimum representative flight duration in a vibrating environment with whole body frequencies similar to those experienced in-flight.

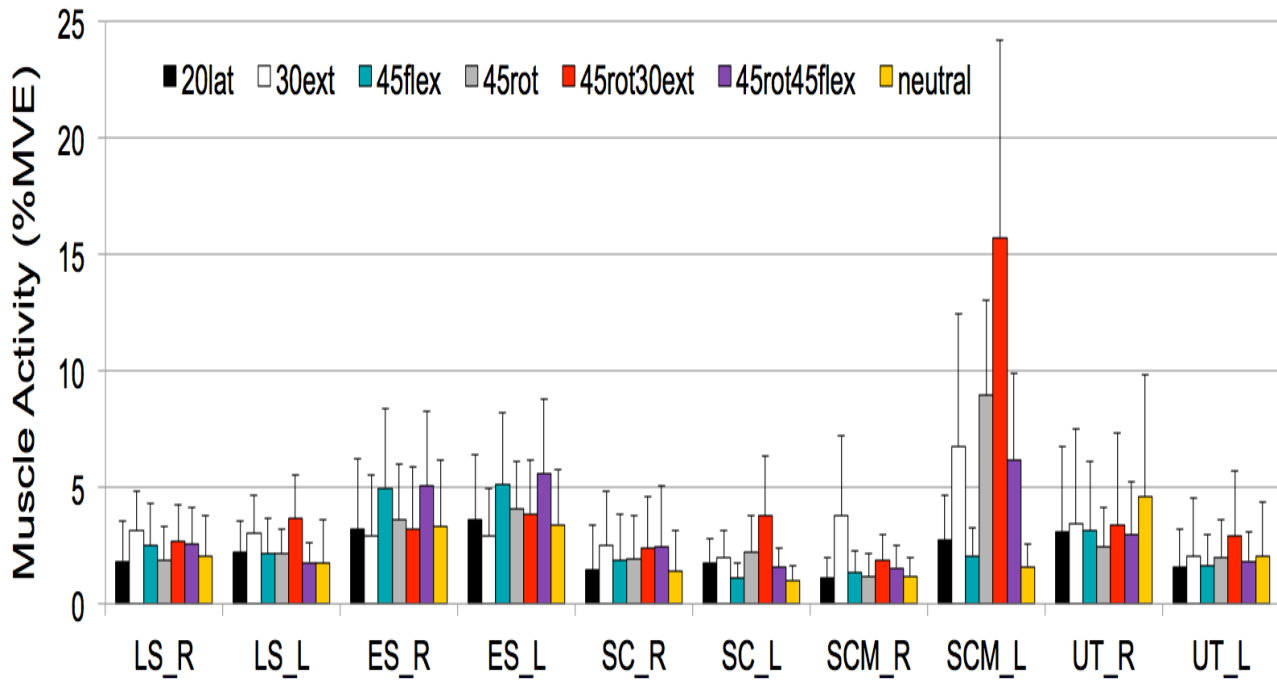


Figure 10: Average peak EMG (+ 1 SD) across participants and helmets illustrating the low magnitudes of muscle activity.

Table 4: The **percentage** of individual trials that exceeded the Jonsson (1978) APDF criterion values for the 10 muscles examined.

Muscle	APDF Percentile Values		
	10	50	90
LS-R	0.2	0.0	0.0
LS-L	0.2	0.0	0.0
ES-R	5.3	0.0	0.0
ES-L	6.8	0.0	0.0
SC-R	0.0	0.0	0.0
SC-L	0.0	0.0	0.0
SCM-R	0.2	0.0	0.0
SCM-L	5.2	4.6	0.0
UT-R	6.3	1.3	0.0
UT-L	2.6	0.0	0.0

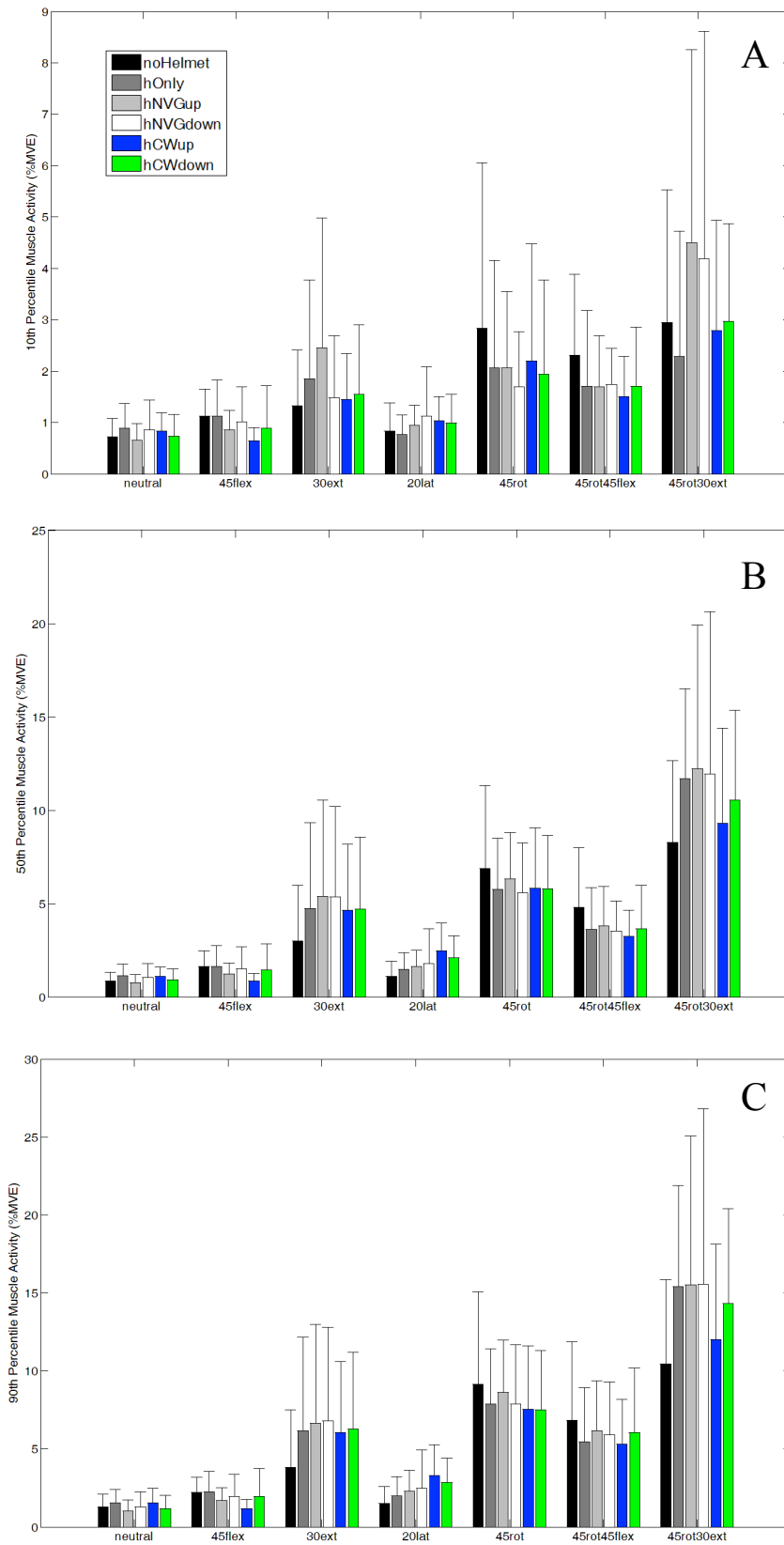


Figure 11: APDF results for the left sternocleidomastoid (SCM-L) for the 10th (A), 50th (B) and 90th (C) percentile criterion measures.

Helmet Main Effects

The six main effect findings, out of 315 comparisons, for the helmet are presented individually. Five of the 6 findings found a difference between the no helmet (baseline control) condition and a helmeted condition; i.e. there was only a single case where responses differed across two helmeted conditions. Further, considering our statistical analysis approach employed an alpha of 0.05, we would expect at least this number of significant findings due to chance alone. Overall, the results indicate that for the slow moving and static hold trials examined in this study, helmet configuration had little influence on the EMG magnitudes recorded.

Mean EMG

The mean EMG of the left levator scapulae (LS-L) demonstrated a significant difference ($p = 0.0136$) between no helmet and the helmet with the night vision goggles attached in the up position (hNVGup). This difference was only present in trial phase 4, where the participant initiates the return from the statically held posture back towards neutral (Figure 12). The difference between the average EMG in this phase for the two conditions was very small with the average value across participants for the no helmet condition being 1.3 %MVE and the hNVGup increasing the value to 1.7 %MVE.

RMS EMG

Similar to the mean EMG finding, only the EMG magnitudes in phase 4 were different for the left levator scapulae (LS-L) ($p = 0.007$) and left splenius capitis (SC-L) ($p = 0.0035$). Both of these muscles only had differences for the no helmeted condition compared to the hNVGup and the helmet with the counter weight attached in the down (hCWdown) positions. The RMS EMG for no helmet in the return movement initiation was 1.5 %MVE compared with 1.7 %MVE and 1.9 %MVE for the hNVGup and hCWdown helmeted conditions respectfully. The addition of these two helmet configurations negligibly increased the muscular demand during movement initiation but had no impact during the hold phase nor did the helmet configurations differ from each other, only from a non-helmets control comparison.

Time to maximum

The time to maximum activity had one significant difference during the second movement phase, the component when the participant was controlling the helmet and stopping their head in the static hold position. The right levator scapulae (LS-R) demonstrated differences between two helmeted conditions with the NVG system attached and set in the down (hNVGdown) positions being different than the helmet with CW attached and NVG attached and down (hCWdown). The addition of the counter weight increased the time to maximum muscle activity of 937 ms and was shorter for the condition with no counter weight attached at 708 ms. While the muscle was activated earlier, likely due to counter the weight of the NVG that were unbalanced by a counter weight, this finding must be considered in the context that while muscles were turned on earlier, there were no differences in the magnitude of activation for this muscle in this or any other trial phase. The finding does indicate some potential for the counter weight to reduce total demand, as if the muscle is turned on longer, even at the same magnitude this could translate to a potential increase in risk. Albeit, given the small magnitudes of activation for this muscle (mean for LS-R in phase 2 for the two helmet conditions were 1.4 %MVE for hNVGdown and 1.6 %MVE for

hCWdown with peak EMG in the same phase of 2.6 %MVE and 2.7 %MVE respectively) this increased time with the muscle active would introduce marginal differences biologically.

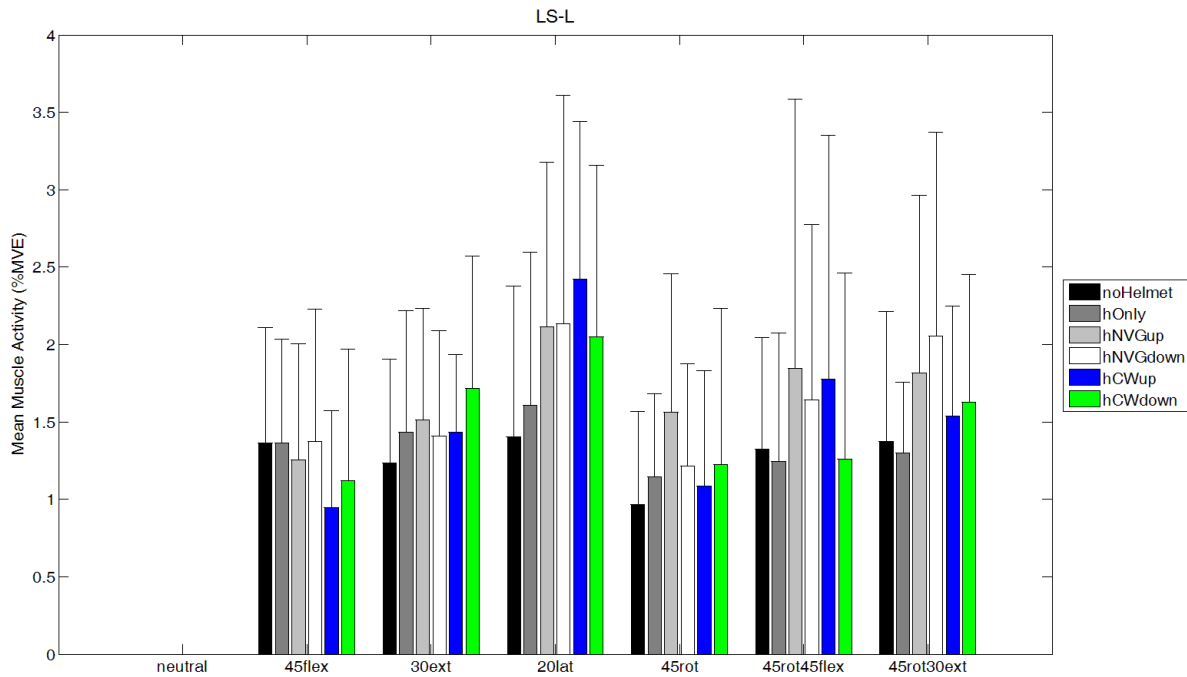


Figure 12: Average EMG magnitudes for the left levator scapulae (LS-L) with the noHelmet (black series) being significantly different from the helmet with the night vision goggles attached in the up position (hNVGup – light gray series).

APDF Results

Similar to the RMS and mean EMG results, when examining the activation profiles over an entire 20 second trial using the amplitude probability distribution function only the left splenius capitis (SC-L) muscle revealed any differences across helmet conditions and only when the no helmeted condition was compared against the helmet with the night vision goggles attached in the up position (hNVGup) ($p = 0.0398$ for the median (50th percentile), and $p = 0.0396$ for peak (90th percentile)) (Figure 13). The addition of the night vision goggles in the up position increased muscular activation compared to wearing no helmet. There were no differences between any of the helmeted conditions and the magnitude of EMG for the 50th and 90th percentile measures were far below the lower bounds of the proposed injury threshold values of 10 %MVE and 50 %MVE, respectively, with average activations all falling below approximately 5 %MVE.

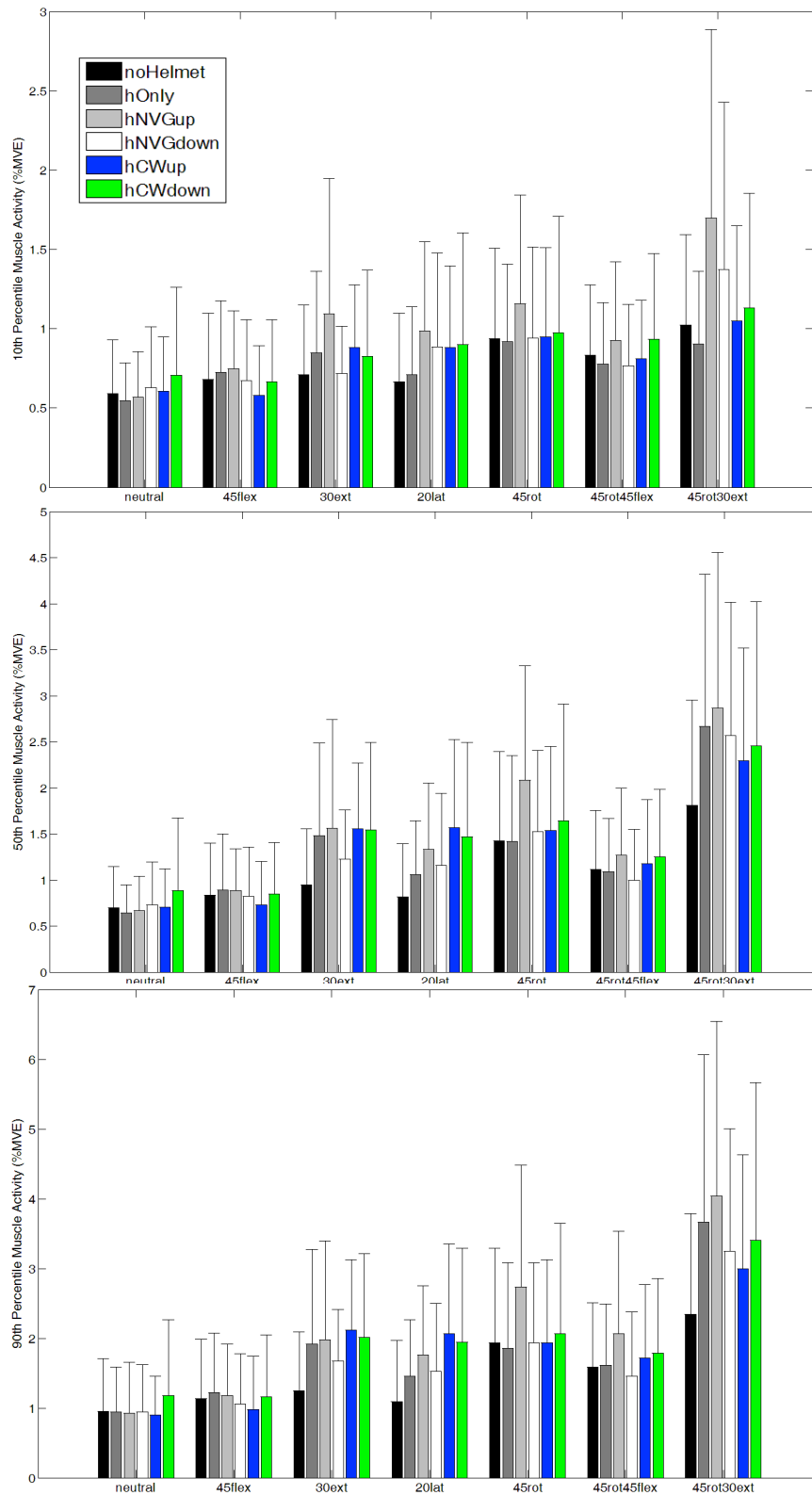


Figure 13: APDF results for the left splenius capitis for the 50th (middle plot) and 90th (lower plot) percentile criterion measures. The noHelmet (black series) was found to be significantly different from the helmet with the night vision goggles attached in the up position (hNVGup – light gray series). Note – the 10th percentile criterion was not a significant main effect but has been included for reference.

Helmet by Posture Interactions

The interaction of helmet and posture, indicating that the helmet condition influenced changing responses depending on the postures that were held, resulted in 83 statistically significant findings (Appendix A5). As indicated in the main effects discussion, overall the muscular demands associated with wearing the helmet, regardless of the configuration were very low. Several examples are provided to illustrate the interactions present. The interactions were largely driven by the helmet responses to postures that aligned with the increasing demands of additional load being added to the helmet or by scenarios where the helmeted conditions responded similarly to each other but different to the control or no helmet condition. Peak EMG has been selected to illustrate the interactions as this measure reflects the maximal demand present, and given the overall low magnitudes of EMG, clearly demonstrates the low demand associated with each of the experimental conditions. Interactions were present for all trial phases, muscles and the majority of variables examined, with the exception of the EMG onset measure that yielded no interaction or helmet main effects.

The most common scenario driving the interaction was that there was a wide range of responses that varied by helmet condition across the postures evaluated. For example the upper plot in figure 14 presents the peak EMG for the static hold phase (phase 3) for the right levator scapulae (LS-R). The changing response of the muscle to helmet condition can be visualized by looking across the trial conditions (x-axis on the figure) and how the helmet conditions respond differently. In this case, the interaction is driven by the two rightmost conditions (45rot45flex and 45rot30ext) where the no helmet has the highest muscular activity in the first and changes to the lowest in the 45rot30ext, whereas the Helmet with night vision goggles down and counter weight attached (hCWdown) exhibits the exact opposite response. A different type of interaction is demonstrated for the right sternocleidomastoid (SCM-R) in the lower plot of figure 14. All of the postures induce similar low level peak muscular activations with the exception of 30° extension, which is a posture that requires SCM involvement to offset the externally created neck extensor moment. In this condition (30ext) the helmet conditions all have a far greater direction changes than the small increase seen in the control no helmet condition.

Neck Muscle Co-activation

The overall activation of the neck musculature was assessed using the Muscular Co-activation Index (MCI) calculation. This measure takes all muscles and combines their activations to gain an overall appreciation of the net activity required during the five phases of the tasks. There were no helmet main effects on this variable in any of the five phases and the sole statistical finding was for a helmet by posture interaction during the last phase, when the participants were stopping the helmet on returning to the neutral posture (Figure 15). The no helmet condition had the highest total activation in the 45flex condition with the two counter weight conditions having the highest value in the combined posture of rotation and extension (45rot30ext).

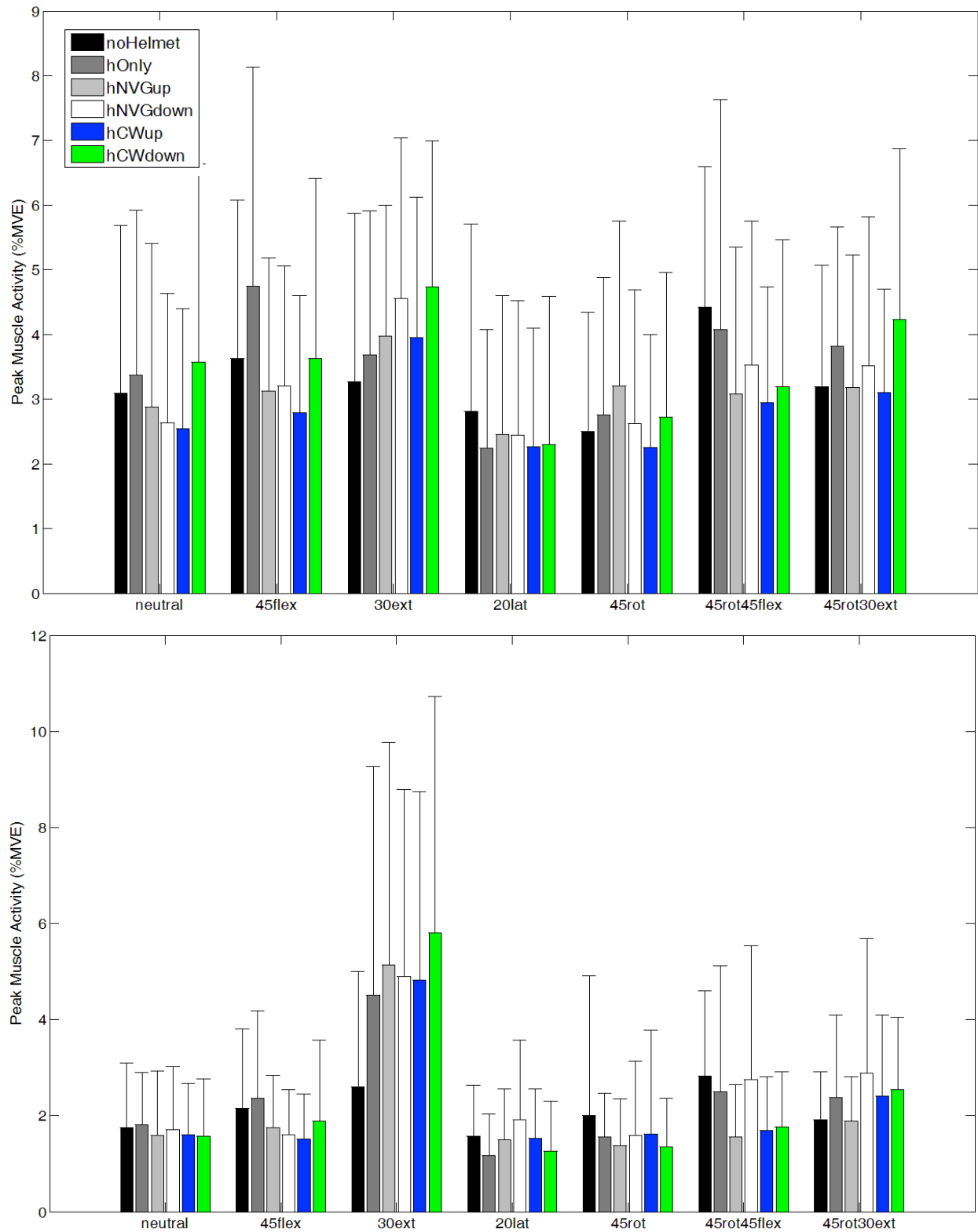


Figure 14: Peak EMG (± 1 SD) demonstrating statistical interactions between helmet and posture during the static hold phase (phase 3) for the right levator scapulae (LS-R upper plot) and right sternocleidomastoid (SCM-R lower plot).

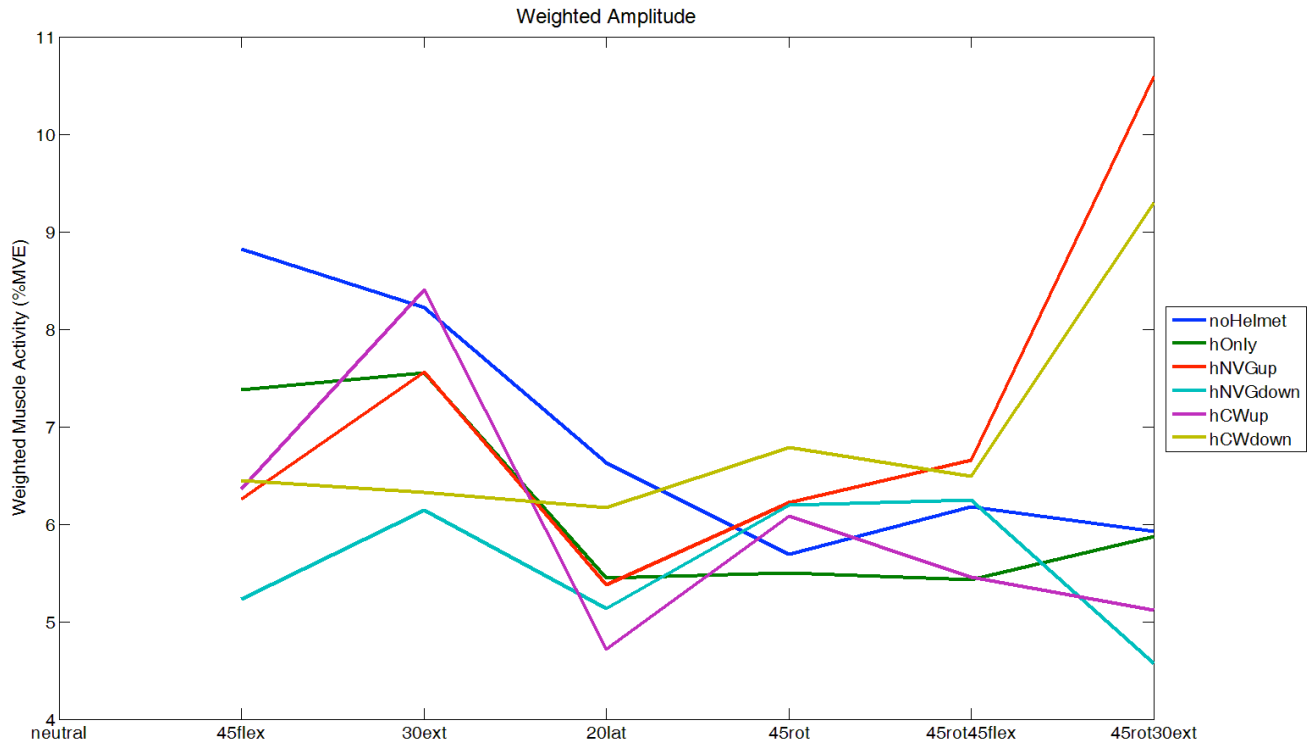


Figure 15: An interaction plot of the fifth trial phase for the co-activation weighted EMG measure.

Given the interaction in phase 5 for the co-activation, the static hold task component (phase 3) was evaluated to get an overall impression of the demands associated with the helmets in the different positions (Figure 16). While there were no significant helmet or interaction (posture by helmet) effects for this phase the weighted co-activation provides insight into the muscular response to the helmet and postural demands. In general terms the addition of the helmet, night vision goggles and counter weight have minimal increase on the muscular response for the majority of postures evaluated in comparison to wearing no helmet. The exceptions to this are the neck postures that required neck extension, where there were increases above wearing no helmet on the order of 3 to 5 %MVE across the 10 muscles. However, when a more appropriate baseline for comparison of the night vision goggles and counter weight is employed, namely wearing the helmet alone, there are no noticeable differences created by the addition of the two additional components.

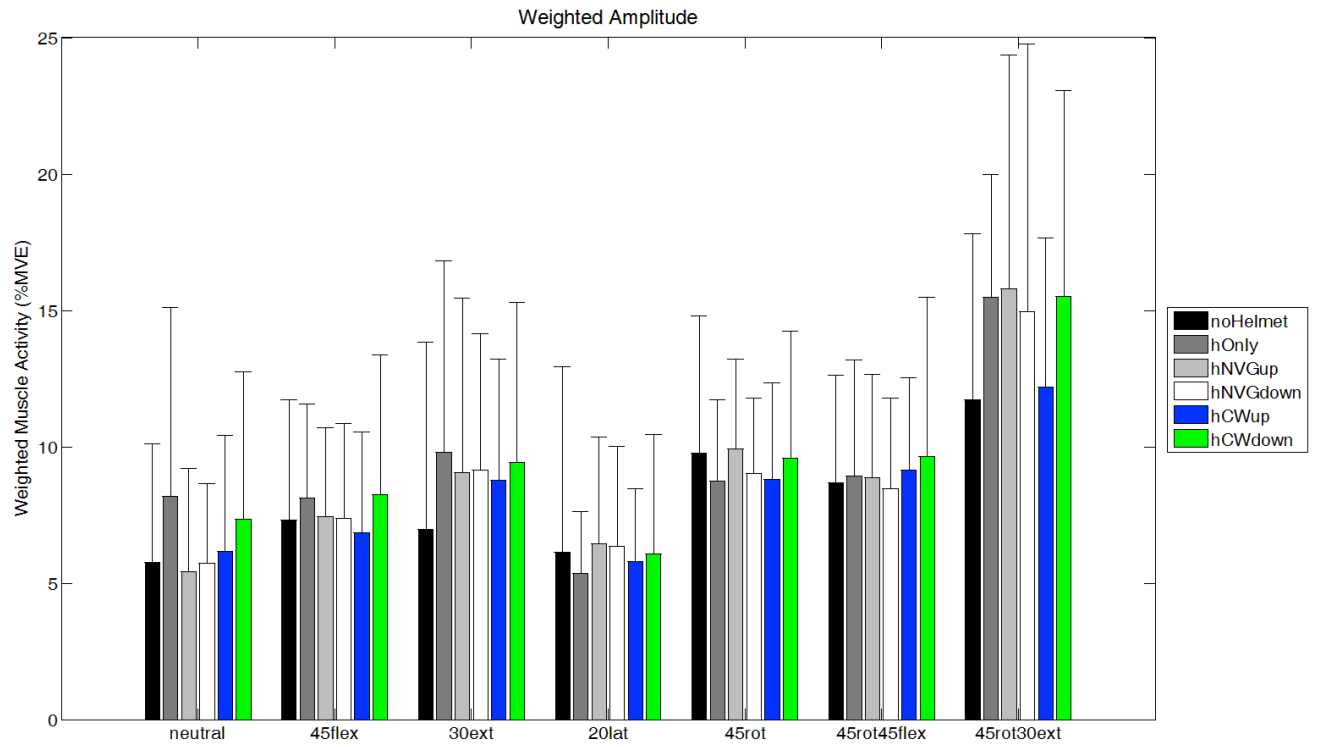


Figure 16: The EMG co-activation weighted measure for the static hold phase of the movement trials.

CONCLUSIONS:

Overall, the wide variety of helmet configurations, which were tested under tightly controlled postural conditions, had minimal influence on the muscular demands during the low velocity movements and static holds examined in this study. The current findings neither support the usage of a counter weight mechanism, nor do they discount its usage. The use of night vision goggles with or without the counter weight did not produce notable changes in muscular responses during the static holds evaluated.

Neck Joint Strength

In the neutral posture, axial rotation and lateral bend had lower joint strength than flexion or extension. The largest posture-related difference observed was for axial rotation conditions, as neck strength increased by 107% for a 45° axial rotation with 45° flexion posture compared to axial rotation in a neutral neck posture. These findings have potential implications for the design of work to avoid these postures, especially in the presence of additional equipment such as helmets that increase joint moments in non-neutral postures. The lowest joint strength was in axial twist, indicating additional loading of the head and neck would have the greatest potential for injury during tasks requiring moments to be created about this axis either in isolation or coupled movements.

Neck Joint Motion (Kinematics)

The end positions across the seven postural tasks were different as intended. However, within each posture there were no kinematic differences across helmet conditions. This indicates that the participants had similar movement strategies in each of the 6 helmet conditions, and any differences are attributable to the helmet configuration and not variability between task performances. Peak angles and total ranges of motion were the same across participants and helmets. No compensatory movements in the neck rotations not being targeted (i.e. “off-axis”) were introduced due to the helmet configurations indicating that the helmet did not introduce any differences in the way participants moved within each postural condition.

Neck Muscular Response to Helmet Configuration

A large number of data processing approaches and outcome variables were employed to assess potential differences in neck muscular responses across the helmet configurations. Overall, muscular demands were modest for all muscles and tasks, ranging between 0.6 to 12.9% MVE for mean EMG. Only 6 main effects of helmet out of 315 statistical comparisons existed. These results demonstrated minimal biological relevance with changes between no helmet and helmeted trials of less than 0.5 %MVE for mean and RMS EMG. When the magnitudes of the APDF were examined for the 10th, 50th, and 90th percentiles criterion values there were no values that exceeded proposed threshold limit values indicating minimal risk for muscle overload for the tasks and helmet conditions studied. There were a large number of interactions between the helmet configurations and the postures adopted, however these existed within the context of the very low magnitudes of muscle activation and often reflected differences attributable to wearing a helmet versus no helmet. Importantly, when the modified helmet conditions were compared to a simple helmet condition as the baseline measure instead of no helmet, differences were negligible.

Considerations

While the muscular responses to the helmet conditions were quite low, they must be contextualized with regard to the experimental conditions examined. The movements in this study were low velocity, based on intentional characterization of muscular demands during the 15-second static holds. The addition of the masses to the helmets in these low acceleration movements had minimal influence on the response of the neck musculature. Future analyses should consider high-fidelity replication of the rate of movements, type of movements and number of movements that occur during in-flight activities. Given the additional mass and resulting inertial changes to the head and neck complex, high acceleration movements would likely have a non-linear influence on muscular activation in both the acceleration to initiate movement as well as the deceleration to stop the rotation of the head, neck and helmet, particularly for tasks that require substantial postural movement.

REFERENCES:

- Aydog, Sedat Tolga, Ertunc Turbedar, A. Haydar Demirel, Onur Tetik, Ahmet Akin, and Mahmut Nedim Doral. 2004. Cervical and Lumbar Spinal Changes Diagnosed in Four-View Radiographs of 732 Military Pilots. *Aviation, Space, and Environmental Medicine* no. 75 (2):154-157.
- Aström, C., Lindkvist, M., Burström, L., Sundelin, G., Karlsson, J.S. 2007. Changes in EMG activity in the upper trapezius muscle due to local vibration exposure. *J Electromyogr Kinesiol.* 19(3):407-15.
- Berg, H. E., G. Berggren, and P. A. Tesch. 1994. "Dynamic neck strength training effect on pain and function." *Arch Phys Med Rehabil* no. 75 (6):661-5.
- Brault, J.R., Siegmund, G.P., Wheeler, J.B. (2000). Cervical muscle response during whiplash: evidence of a lengthening muscle contraction. *Clinical Biomechanics* 15(6): 426-435.
- Bridger, R.S., Groom, M.R., Jones, H. et al. (2002) Task and postural factors are related to back pain in helicopter pilots. *Aviation, Space and Environmental Medicine*, 73 (8), 805–811
- De Loose, V., M. Van Den Oord, F. Burnotte, D. Van Tiggelen, V. Stevens, B. Cagnie, E. Witvrouw, and L. Danneels. 2008. Individual, work-, and flight-related issues in F-16 pilots reporting neck pain. *Aviation Space and Environmental Medicine* no. 79 (8):779. doi: 10.3357/ase.2285.2008.
- Drake JDM, Callaghan JP. Elimination of electrocardiogram contamination from electromyogram signals: an evaluation of currently used removal techniques. *Journal of Electromyography and Kinesiology* 2006; 16(2):175–87.
- Forde, Kelsey A., Wayne J. Albert, Michael F. Harrison, J. Patrick Neary, James Croll, and Jack P. Callaghan. 2011. Neck loads and posture exposure of helicopter pilots during simulated day and night flights. *International Journal of Industrial Ergonomics* no. 41 (2):128-135. doi: <http://dx.doi.org/10.1016/j.ergon.2011.01.001>.
- Fraser, 2011. Waterloo Contract on EMG Data Collection, Contract no. W7711-123456/001/Tor, DRDC Toronto, Toronto, Canada, unpublished Statement of Work.
- Hämäläinen, O., H. Vanharanta, and T. Kuusela. 1993. Degeneration of cervical intervertebral disks in fighter pilots frequently exposed to high +Gz forces. *Aviation, Space, and Environmental Medicine* no. 64 (8):692-696.
- Harms-Ringdahl, Karin, and Kristina Schüldt. 1989. "Maximum neck extension strength and relative neck muscular load in different cervical spine positions." *Clinical Biomechanics* no. 4 (1):17-24. doi: [http://dx.doi.org/10.1016/0268-0033\(89\)90063-6](http://dx.doi.org/10.1016/0268-0033(89)90063-6).

- Hendriksen, I. J., and M. Holewijn. 1999. Degenerative changes of the spine of fighter pilots of the Royal Netherlands Air Force (RNLAf). *Aviat Space Environ Med* no. 70 (11):1057-63.
- Jennum, Poul, Jakob Kjellberg, Rikke Ibsen, and Tom Bendix. 2013. Health, Social, and Economic Consequences of Neck Injuries: A Controlled National Study Evaluating Societal Effects on Patients and Their Partners. *Spine* no. 38 (5):449-457
10.1097/BRS.0b013e3182819203.
- Jonsson, B., 1978. Kinesiology: with special reference to electromyographic kinesiology. In: *Contemporary clinical neurophysiology*, edited by W. A. Cobb and H. Duija, Amsterdam: Elsevier Science Publishers.
- Jordan, Alan, Jesper Mehlsen, Per Martin Bulow, Keld Ostergaard, and Bente Danneskiold-Samsoe. 1999. "Maximal Isometric Strength of the Cervical Musculature in 100 Healthy Volunteers." *Spine* no. 24 (13):1343.
- Lange, B., J. Torp-Svendsen, and P. Toft. 2011. Neck pain among fighter pilots after the introduction of the JHMCS helmet and NVG in their environment. *Aviation Space and Environmental Medicine* no. 82 (5):559. doi: 10.3357/ase.2935.2011.
- Laursen, B., Jensen, B.R., Garde, A.H., Jørgensen, A.H. 2002. Effect of mental and physical demands on muscular activity during the use of a computer mouse and a keyboard. *Scandinavian Journal of Work, Environment & Health*. 28(4): 215-221.
- Leinonen, V., Kankaanpää, M., Vanharanta, H., Airaksinen, O., Hänninen, O. 2005. Back and neck extensor loading and back pain provocation in urban bus drivers with and without low back pain. *Pathophysiology*. 12(4):249-55.
- Lu WW, Bishop PJ. Electromyographic activity of the cervical musculature during dynamic lateral bending. *Spine* 1996 Nov 1;21(21):2443-9.
- MacKinnon, Colum D., and David A. Winter. 1993. Control of whole body balance in the frontal plane during human walking. *Journal of Biomechanics* no. 26 (6):633-644. doi: [http://dx.doi.org/10.1016/0021-9290\(93\)90027-C](http://dx.doi.org/10.1016/0021-9290(93)90027-C).
- Magnusson, M.L., Pope, M.H., Hasselquist, L. (1999). Cervical electromyographic activity during low-speed rear impact. *European Spine Journal* 8(2): 118-125.
- Mayoux-Benhamou, M. A., and M. Revel. 1993. "Influence of head position on dorsal neck muscle efficiency." *Electromyography and clinical neurophysiology* no. 33 (3):161-166.
- Minasian, SM, Baklavadzhian, OG, Saakian, SG. 1989. The effect of whole-body vibration on the electrical activity and oxidative metabolism in different brain structures. *Gig Tr Prof Zabol.* (12):22-6.

- Moroney, Sean P., Albert B. Schultz, and James A. A. Miller. 1988. "Analysis and measurement of neck loads." *Journal of Orthopaedic Research* no. 6 (5):713-720. doi: 10.1002/jor.1100060514.
- Nakashima, 2014. Analysis of electromyography (EMG) and motion capture (MOCAP) data for aircrew neck strain, Canadian Institute for Military and Veterans Health Research (CIMVHR) Task 28, Defence Research and Development Canada, Toronto Research Centre, unpublished Statement of Work.
- Petrén-Mallmin, M., and J. Linder. 1999. MRI cervical spine findings in asymptomatic fighter pilots. *Aviation, Space, and Environmental Medicine* no. 70 (12):1183-1188.
- Phillips, C. A., Petrofsky, J. S. Neck muscle loading and fatigue: Systematic variation of headgear weight and center-of-gravity. *Aviation, Space and Environmental Medicine* 54:901-905, 1983
- Pippig, T., and J. Kriebel. 2000. Prevalence of cervical and lumbar disc disorders in pilots of the German armed forces. *European journal of medical research* no. 5 (1):5-8.
- Queisser, F., R. Blüthner, and H. Seidel. 1994. "Control of positioning the cervical spine and its application to measuring extensor strength." *Clinical Biomechanics* no. 9 (3):157-161. doi: [http://dx.doi.org/10.1016/0268-0033\(94\)90015-9](http://dx.doi.org/10.1016/0268-0033(94)90015-9).
- Rietdyk, S., Patla, A. E., Winter, D. A., Ishac, M. G., & Little, C. E. (1999). NACOB presentation CSB New Investigator Award. Balance recovery from medio-lateral perturbations of the upper body during standing. *Journal of Biomechanics*, 32(11), 1149–1158.
- Sacher, N., Frayne, R.J., Dickey, J.P. (2012). Investigating cervical muscle response and head kinematics during right, left, frontal and rear-seated perturbations. *Traffic Injury Prevention* 13 (5): 529-536.
- Salmon, Danielle M., Michael F. Harrison, and J. Patrick Neary. 2011. Neck Pain in Military Helicopter Aircrew and the Role of Exercise Therapy. *Aviation, Space, and Environmental Medicine* no. 82 (10):978-987. doi: 10.3357/ase.2841.2011.
- Schall, D. G. 1989. Non-ejection cervical spine injuries due to +Gz in high performance aircraft. *Aviat Space Environ Med* no. 60 (5):445-56.
- Schleifer, L.M., Spalding, T.W., Kerick, S.E., Cram, J.R., Ley, R., Hatfield, B.D. 2008. Mental stress and trapezius muscle activation under psychomotor challenge: A focus on EMG gaps during computer work. *Psychophysiology* 45(3): 356–365. DOI: 10.1111/j.1469-8986.2008.00645.x
- Seng, Kok-Yong, Vee-Sin Lee Peter, and Pin-Min Lam. 2002. "Neck muscle strength across the sagittal and coronal planes: an isometric study." *Clinical Biomechanics* no. 17 (7):545. doi: 10.1016/s0268-0033(02)00067-0.

- Staudte, H. W., and N. Dühr. 1994. "Age- and sex-dependent force-related function of the cervical spine." *European Spine Journal* no. 3 (3):155-161. doi: 10.1007/bf02190578.
- Strimpakos, Nikolaos, Vasiliki Sakellari, Georgios Gioftsos, and Jacqueline Oldham. 2004. "Intratester and intertester reliability of neck isometric dynamometry." *Arch Phys Med Rehabil* no. 85 (8):1309-1316. doi: <http://dx.doi.org/10.1016/j.apmr.2003.08.104>.
- Van Den Oord, M. H. A. H., V. De Loose, T. Meeuwsen, J. K. Sluiter, and M. H. W. Frings-Dresen. 2010. Neck pain in military helicopter pilots: Prevalence and associated factors. *Military medicine* no. 175 (1):55.
- Vanderbeek, R. D. 1988. Period prevalence of acute neck injury in U.S. Air Force pilots exposed to high G forces. *Aviat Space Environ Med* no. 59 (12):1176-80.
- Vasavada, Anita N., Siping Li, and Scott L. Delp. 2001. "Three-Dimensional Isometric Strength of Neck Muscles in Humans." *Spine* no. 26 (17):1904-1909.
- Wu, G., Siegler, S., Allard, P., Kirtley, C., Leardini, A., Rosenbaum, D., Whittle, M., D'Lima, DD., Cristofolini, L., Witte, H., Schmid, O., Stokes, I.; Standardization and Terminology Committee of the International Society of Biomechanics. ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion--part I: ankle, hip, and spine. *Journal of Biomechanics*. 2002 Apr;35(4):543-8.
- Wu, G., van der Helm, F.C., Veeger, H.E., Makhsous, M., Van Roy, P., Anglin, C., Nagels, J. and others. ISB Recommendation on Definitions of Joint Coordinate Systems of Various Joints for the Reporting of Human Joint Motion--Part II: Shoulder, Elbow, Wrist and Hand. *Journal of Biomechanics* 38, no. 5 (2005): 981-992.
- Ylinen, Jari J., Asghar Rezasoltani, Mikko V. Julin, Hilkka A. Virtapohja, and Esko A. Mätkiä. 1999. "Reproducibility of isometric strength: measurement of neck muscles." *Clinical Biomechanics* no. 14 (3):217-219. doi: [http://dx.doi.org/10.1016/S0268-0033\(98\)00063-1](http://dx.doi.org/10.1016/S0268-0033(98)00063-1).
- Ylinen, Jari, Petri Salo, Matti Nykänen, Hannu Kautiainen, and Arja Häkkinen. 2004. "Decreased isometric neck strength in women with chronic neck pain and the repeatability of neck strength measurements." *Arch Phys Med Rehabil* no. 85 (8):1303-1308. doi: <http://dx.doi.org/10.1016/j.apmr.2003.09.018>.

APPENDICES:

Appendix A1: Data file directory listing and MATLAB file structure descriptions

1.0 DATA FILE DIRECTORY

\DATA PROCESSING

\Administrative [*Documents used for experimental collection sessions*]

- File Directory (.pdf) – THIS FILE

\Analysis [*Data analysis files for EMG, neck moment, and neck motion/angle data. Function descriptions are located in the header of each individual m-file.*]

\EMG

- data_pad.m
- emg_linear_envelope.m
- emg_mve_summarize.m
- emg_normalize.m
- emg_plot.m
- emg_plot_onset.m
- EMG_PROCESS.m
- emg_trial_summarize.m
- file_characteristics.m
- file_loader.m
- fix_outputs.m
- track_progress.m

\MOMENT

- load_neutral.m
- moment_convert_to_neckCS.m
- moment_filter_force.m
- moment_filter_posture.m
- moment_mve_corrections.m
- MOMENT_PROCESS.m
- track_progress.m

\MOTION

- calculate_lcs.m
- data_pad.m
- file_characteristics.m
- file_loader.m
- motion_head_angle.m
- motion_L5reconstruct.m
- motion_phases.m
- motion_plot.m
- MOTION_PROCESS.m
- motion_summarize.m
- normr.m
- track_progress.m

\Data

\Moment

\Corrections Factors

-CorrectionsFactors_MVE (.xlsx)

\Helmet-Cube Distances

-helmetcube_fe_mount (.xlsx)

-helmetcube_lat_mount (.xlsx)

-helmetcube_rot_mount (.xlsx)

\Transformations

-RotationTranslationMatrix (.mat)

\Output

-Summary variables for EMG, motion, and moment data (.xlsx)

-Summary tables containing means and standard deviations (.xlsx)

\UW002-UW010

\Force-Moment Global

-Time-varying XYZ neck forces and moments, in global (.csv)

[Fx Fy Fz Mx My Mz]

\Force-Moment Local (Neck)

-Time-varying XYZ neck forces and moments, in local (.csv)

[Fx Fy Fz Mx My Mz]

\Neck Angle

-Time-varying XYZ neck and moments, in global (.csv)

[x-angle y-angle z-angle]

\Processed EMG

-Time-varying 10-muscle EMG, in volts (MVE) or %MVE (trials)

[LS-R LS-L ES-R ES-L SC-R SC-L SCM-R SCM-L UT-R UT-L]

\Raw

\EXP

-Experimental trials (.csv)

\MVE

-MVE trials (.csv)

-neutral (.csv)

-neutral_hat (.csv)

-standing (.csv)

-standing_hat (.csv)

-UW0**correct (.mat)

-UW0**endpoints (.mat)

-UW0**masks (.mat)

-mve_summary (.xlsx)

-Reference Codes (.xlsx)

2.0 MATLAB FILE HEIRARCHY

The following is a basic hierarchy of the MATLAB m-files used for data processing. Data is processed in three separate streams according to the three types of outputs variables: EMG, MOMENT, and MOTION. Data processing for each stream of data is initiated from the capitalized m-file within the associated analysis subfolder.

2.1 EMG DATA PROCESSING

Electromyography (EMG) data is processed by initiating EMG_PROCESS.m within the EMG analysis subfolder. This process calculates peak, mean, median, RMS, weighted, and APDF values for amplitude, and time to maximum and time to onset of activation. These values are calculated for each of the 5 movement phases and output into the appropriate summary files (summary_emg_*.xlsx).

>EMG_PROCESS

- >file_loader (selects files to process)
- >emg_linear_envelope (applies heart rate filter and linear envelope to EMG data)
 - >data_pad (adds padding points to data prior to processing, then removes)
- >file_characteristics (isolates trial number, helmet, and posture from filename)
- >emg_normalize (normalize EMG data to MVE trials)
- >emg_plot (plot EMG data)
- >emg_trial_summarize (calculate EMG data summary variables)
 - >fix_outputs (adjust data for absent phases)
 - >emg_plot_onset (plot muscle activity onset points on EMG plots)
- >track_progress (indicate processing status)
- >emg_mve_summarize (find maximum amplitude across MVE trials for each muscle)

2.2 MOMENT DATA PROCESSING

Neck moment data is processed by initiating `MOMENT_PROCESS.m` within the `MOMENT` analysis subfolder. This process calculates peak XYZ neck forces and moments and outputs into the appropriate summary file (`summary_motion_forcemoment.xlsx`).

>MOMENT_PROCESS

- >moment_mve_corrections (calculate correction factors for force and moment)
- >load_neutral (load posture from a neutral trial)
- >moment_filter_force (filter force and moment data)
- >moment_filter_posture (filter posture data)
- >moment_convert_to_neckCS (calculate local neck force and moment)
- >track_progress (indicate processing status)

2.3 MOTION DATA PROCESSING

Motion data is processed by initiating `MOTION_PROCESS.m` within the `MOTION` analysis subfolder. This process calculates XYZ neck angles, and outputs peak neck angle and neck angle range of motion for each of the 5 movement phases into the appropriate summary files (`summary_motion_*.xlsx`).

>MOTION_PROCESS

- >file_loader (selects files to process)
- >motion_L5reconstruct (calculates L5 location vector & neutral direction cosine matrix)
 - >calculate_lcs (calculates local coordinate system for the head and thorax)
- >file_characteristics (isolates trial number, helmet, and posture from filename)
- >motion_head_angle (calculates neck angles—head relative to thorax)
 - >calculate_lcs (calculates local coordinate system for the head and thorax)
 - >data_pad (adds padding points to data prior to processing, then removes)
 - >motion_plot (plots neck angle and neck angular velocity)
- >motion_phases (calculates motion phases endpoints using angular velocity threshold)
 - >motion_plot (plots motion phase endpoints)
- >motion_summarize (calculate motion data summary variables)
- >track_progress (indicate processing status)

Appendix A2: MATLAB code for processing kinetic (Moment) data.

```

function MOMENT_PROCESS(calcCorr,outData)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION:
%Reads in kinetics (force and moment) data files, and calculates
%time-series neck reaction forces and moment (XYZ) in the neck local
%system. Peak force and moment values are exported to an Excel spreadsheet.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
tstart = tic;

cd ../..
info.root = [cd '\Data\'];
cd('Analysis\MOMENT')
info.subjects = [{'UW002'},{'UW003'},{'UW005'},{'UW006'},{'UW007'},{'UW008'},↵
{'UW009'},{'UW010'}]; %Participant Identification Codes

info.fs_force = 1500; %Force data sample frequency (Hz)
info.fc_force = 4.0; %Force data filter cutoff frequency (Hz)

%Rotation Sequence
Rot = [0 0 1;1 0 0;0 1 0];

if calcCorr == 1
    %Calculate MVE Force and Moment Correction Factors *****
    fprintf('Calculating MVE Correction Factors...\n')
    [H_C,H_g,F_Corr] = moment_mve_corrections(info,Rot);
    toc(tstart)
end

%Import Participant Data and Shunt Conversion Key *****
[~,~,shunt_key] = xlsread([info.root 'Moment\force cube conversion key.↵
xlsx'],'Sheet1','A1:AE9');

for M = 1:size(info.subjects,2) %SUBJECTS
    %Load MVE trial file names *****
    subid = info.subjects{M}; %UW002
    filenames = dir([info.root info.subjects{M} '\Raw\MVE\mvc*.csv']);
    filename = {filenames.name};
    numfiles = size(filename,2);

    %Identify row for shunt conversion key
    for j = 1:size(shunt_key,1)
        if strcmp(info.subjects{M},shunt_key(j,1))
            shuntrow = j;
            break
        end
    end

    fprintf('Loading Neutral Posture Files...\n')
    %Load Neutral Trial Postures
    [nFE,nLAT,nROT] = load_neutral(info,info.subjects{M},Rot);

```

```

for N = 1:numfiles %TRIALS
    if N == 1
        fprintf('\nProcessing Trial Data:\n')
    end

    %Determine Trial Type
    trialtype{N} = filename{N}(6:end-4);
    trialnum{N} = filename{N}(1:4);
    subject{N} = info.subjects{M};

    %Import Force Data *****
    headerRow = 5; %Number of header rows in data files
    headerCol = 2; %Number of header columns in data files
    imdata = importdata([info.root subject{N} '\Raw\MVE\' filename{N}],',',↵
headerRow); %import
    forcedata = imdata.data(:,headerCol+11:end);

    %Import Posture Data *****
    headerRow = 10511;
    imdata = importdata([info.root subject{N} '\Raw\MVE\' filename{N}],',',↵
headerRow); %import
    posturedata = imdata.data(:,headerCol+1:end);

    %Filter Force and Moment Data and Covert Units *****
    force_filt = moment_filter_force(info,forcedata,shunt_key,shuntrow);

    if outData == 1
        %Write Filtered Force Data, in Newtons *****
        dlmwrite([info.root subject{N} '\Force-Moment Global\' filename{N}(1:end-↵
4) '_forceG.csv'],force_filt); %Force Data, in Newtons
    end

    %Rotate, Pad, and Filter Posture Data
    posturedata = moment_filter_posture(info,posturedata,Rot);

    %Convert Forces/Moments to Neck Coordinate System
    [neck_moment{N},neck_angles{N}] = moment_convert_to_neckCS(filename{N},info,↵
subject{N},force_filt,posturedata,trialtype{N},nFE,nLAT,nROT,H_C,H_g,F_Corr,outData);
    track_progress(N)
end

%Write FORCE/MOMENT SUMMARY DATA to Files
xlswrite([info.root 'Output\summary_kinetic_forcemoment.xlsx'],[subject' trialnum'↵
trialtype'],info.subjects{M},'A3');
xlswrite([info.root 'Output\summary_kinetic_forcemoment.xlsx'],cell2mat↵
(neck_moment'),info.subjects{M},'D3');
% xlswrite([info.root 'Output\kinetic_summary_angles.xlsx'],[subject' trialnum'↵
trialtype'],info.subjects{M},'A3');
% xlswrite([info.root 'Output\kinetic_summary_angles.xlsx'],cell2mat↵
(neck_angles'),info.subjects{M},'D3');

clear subject trialnum trialtype neck_moment neck_angles;

fprintf([info.subjects{M} 'Complete.\n'])

```

```
    toc(tstart)
end
```

```

function [H_C,H_g,F_Corr] = moment_mve_corrections(info,Rot)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, Caroline Joseph, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION:
%Calculates rotation matrices, translations matrices, and correction
%factors for force and moment values based on baseline force and moment
%trials.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% FE MOUNT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Centre of Gravity (CG) Calculations
FxFE = 1.364984983;
FyFE = -20.28482546;
FzFE = 1.530432579;
MxFE = -4.108119218;
MyFE = -0.040179728;
MzFE = 0.535443201;
F_FE = [FxFE;FyFE;FzFE];

AFEcg = [0,FzFE,FyFE;FzFE,0,FxFE;FyFE,FxFE,0];
M_FE = [MxFE;MyFE;MzFE];
cg_FE = AFEcg\M_FE; %--> cg position in force cube CS in m

% ALatcg = [0,FzLat,FyLat;FzLat,0,FxLat;FyLat,FxLat,0];
% M_Lat = [MxLat;MyLat;MzLat];
% cg_Lat = ALatcg\M_Lat; %--> cg position in force cube CS in m
%
% ARotcg = [0,FzRot,FyRot;FzRot,0,FxRot;FyRot,FxRot,0];
% M_Rot = [MxRot;MyRot;MzRot];
% cg_Rot = ARotcg\M_Rot; %--> cg position in force cube CS in m

%Import data for original positioning of mount
FileNameFE = [info.root 'Moment\Helmet-Cube Distances\helmetcube_fe_mount.xlsx'];
SheetFE = 'helmetcube_fe_mount';
[numFE,~,~] = xlsread(FileNameFE,SheetFE,'c271:a1271');

%Rotate each marker trajectory according to "Rot"
for i = 1:(size(numFE,2)/3)
    numFE(1,(3*i-2:3*i)) = numFE(1,(3*i-2:3*i))*Rot;
end

%Helmet Coordinate System (H)
kH_FE = (numFE(1,28:30)'\-numFE(1,25:27)')/norm(numFE(1,28:30)'\-numFE(1,25:27)');
iH_FE = cross((numFE(1,31:33)'\-numFE(1,25:27)'),kH_FE)/norm(cross((numFE(1,31:33)'\-
numFE(1,25:27)'),kH_FE));
jH_FE = cross(kH_FE,iH_FE)/norm(cross(kH_FE,iH_FE));
r_g_H_FE = [iH_FE jH_FE kH_FE];
r_H_g_FE = r_g_H_FE';
Ori_H_g_FE = numFE(1,25:27)';

```

```

%Force Cube Coordinate System (C)
iC_FE = (numFE(1,13:15) '-numFE(1,1:3)') / norm(numFE(1,13:15) '-numFE(1,1:3)');
kC_FE = cross(iC_FE, (numFE(1,22:24) '-numFE(1,1:3)')) / norm(cross(iC_FE, (numFE(1,22:24) '-numFE(1,1:3)')));
jC_FE = cross(kC_FE, iC_FE) / norm(cross(kC_FE, iC_FE));
r_g_C_FE = [iC_FE jC_FE kC_FE];
r_C_g_FE = r_g_C_FE';
Ori_C_C_FE = [0.0539206; 0.0612142; 3.3667];

%Determine origin of force cube coordinate system in global
%Point intersecting 3 places (2 diagonals of force cube and front surface)
%plane1--> norm=((f2-f4)x(b2-f4)
Norm1FE = cross((numFE(1,7:9) '-numFE(1,13:15)'), (numFE(1,10:12) '-numFE(1,13:15)')) / norm(cross((numFE(1,7:9) '-numFE(1,13:15)'), (numFE(1,10:12) '-numFE(1,13:15)')));
BFE(1,1) = numFE(1,13:15) * Norm1FE;
%plane2--> norm = ((f1-f3)x(b3-f3)
Norm2FE = cross((numFE(1,22:24) '-numFE(1,1:3)'), (numFE(1,4:6) '-numFE(1,1:3)')) / norm(cross((numFE(1,22:24) '-numFE(1,1:3)'), (numFE(1,4:6) '-numFE(1,1:3)')));
BFE(2,1) = numFE(1,1:3) * Norm2FE;
%plane3--> norm = kC_FE
Norm3FE = kC_FE;
BFE(3,1) = numFE(1,1:3) * Norm3FE;
AFE = [Norm1FE'; Norm2FE'; Norm3FE'];

Ori_C_g_FE = AFE \ BFE + r_g_C_FE * Ori_C_C_FE;

%Position and orientation of Force Cube CS in Helmet CS
r_H_C_FE = r_H_g_FE * r_g_C_FE;
r_C_H_FE = r_H_C_FE';
t_H_C_FE = r_H_g_FE * (Ori_C_g_FE - Ori_H_g_FE);

for N = 1:3

    %Force correction for Neutral Flexion *****
    FileNameNFlex = [info.root 'UW010\Raw\MVE\mvc' num2str(N) '-neutralflex.CSV'];
    NeutralFlex = csvread(FileNameNFlex, 10511, 0);
    NeutralFlex = NeutralFlex(10, 3:end);

    for i = 1:(size(NeutralFlex, 2) / 3)
        NeutralFlex(1, (3*i-2:3*i)) = NeutralFlex(1, (3*i-2:3*i)) * Rot;
    end

    %Helmet Coordinate System (H)
    kH_NF = (NeutralFlex(1, 4:6) '-NeutralFlex(1, 1:3)') / norm(NeutralFlex(1, 4:6) '-NeutralFlex(1, 1:3)');
    iH_NF = cross((NeutralFlex(1, 7:9) '-NeutralFlex(1, 1:3)'), kH_NF) / norm(cross((NeutralFlex(1, 7:9) '-NeutralFlex(1, 1:3)'), kH_NF));
    jH_NF = cross(kH_NF, iH_NF) / norm(cross(kH_NF, iH_NF));
    r_g_nH_NF{N} = [iH_NF jH_NF kH_NF];

    r_H_nH_NF = r_H_g_FE * r_g_nH_NF{N};
    r_nH_H_NF = r_H_nH_NF';

    F_NF{N} = r_C_H_FE * r_H_nH_NF' * r_H_C_FE * F_FE;
    M_NF{N} = [0, F_NF{N}(3, 1), F_NF{N}(2, 1); F_NF{N}(3, 1), 0, F_NF{N}(1, 1); F_NF{N}(2, 1), F_NF{N}(1, 1), 0];
end

```

```
F_NE{N}(1,1),0]*cg_FE;
```

```
%Force correction for Neutral Extension *****
```

```
FileNameNEExt = [info.root 'UW010\Raw\MVE\mvc' num2str(N) '-neutralext.CSV'];
```

```
NeutralExt = csvread(FileNameNEExt,10511,0);
```

```
NeutralExt = NeutralExt(10,3:end);
```

```
for i = 1:(size(NeutralExt,2)/3)
```

```
    NeutralExt(1,(3*i-2:3*i)) = NeutralExt(1,(3*i-2:3*i))*Rot;
```

```
end
```

```
%Helmet Coordinate System (H)
```

```
    kH_NE = (NeutralExt(1,4:6) '-NeutralExt(1,1:3)')/norm(NeutralExt(1,4:6) '-NeutralExt(1,1:3)');
```

```
    iH_NE = cross((NeutralExt(1,7:9) '-NeutralExt(1,1:3)'), kH_NE)/norm(cross((NeutralExt(1,7:9) '-NeutralExt(1,1:3)'), kH_NE));
```

```
    jH_NE = cross(kH_NE, iH_NE)/norm(cross(kH_NE, iH_NE));
```

```
    r_g_nH_NE{N} = [iH_NE jH_NE kH_NE];
```

```
    Ori_nH_g = NeutralExt(11:3)';
```

```
    r_H_nH_NE = r_H_g_FE*r_g_nH_NE{N};
```

```
    r_nH_H_NE = r_H_nH_NE';
```

```
    F_NE{N} = r_C_H_FE*r_H_nH_NE'*r_H_C_FE'*F_FE;
```

```
    M_NE{N} = [0,F_NE{N}(3,1),F_NE{N}(2,1);F_NE{N}(3,1),0,F_NE{N}(1,1);F_NE{N}(2,1),F_NE{N}(1,1),0]*cg_FE;
```

```
%Force correction for 45flex *****
```

```
FileNameFlex45 = [info.root 'UW010\Raw\MVE\mvc' num2str(N) '-45flex.CSV'];
```

```
Flex45 = csvread(FileNameFlex45,10511,0);
```

```
Flex45 = Flex45(10,3:end);
```

```
for i=1:(size(Flex45,2)/3)
```

```
    Flex45(1,(3*i-2:3*i))=Flex45(1,(3*i-2:3*i))*Rot;
```

```
end
```

```
%Helmet Coordinate System (H)
```

```
    kH_F45 = (Flex45(1,4:6) '-Flex45(1,1:3)')/norm(Flex45(1,4:6) '-Flex45(1,1:3)');
```

```
    iH_F45 = cross((Flex45(1,7:9) '-Flex45(1,1:3)'), kH_F45)/norm(cross((Flex45(1,7:9) '-Flex45(1,1:3)'), kH_F45));
```

```
    jH_F45 = cross(kH_F45, iH_F45)/norm(cross(kH_F45, iH_F45));
```

```
    r_g_nH_F45 = [iH_F45 jH_F45 kH_F45];
```

```
    r_H_nH_F45 = r_H_g_FE*r_g_nH_F45;
```

```
    r_nH_H_F45 = r_H_nH_F45';
```

```
    F_F45{N} = r_C_H_FE*r_H_nH_F45'*r_C_H_FE'*F_FE;
```

```
    M_F45{N} = [0,F_F45{N}(3,1),F_F45{N}(2,1);F_F45{N}(3,1),0,F_F45{N}(1,1);F_F45{N}(2,1),F_F45{N}(1,1),0]*cg_FE;
```

```
%Force correction for 30ext *****
```

```
FileNameExt30 = [info.root 'UW010\Raw\MVE\mvc' num2str(N) '-30ext.CSV'];
```

```
Ext30 = csvread(FileNameExt30,10511,0);
```

```
Ext30 = Ext30(10,3:end);
```



```

for i = 1:(size(Ext30,2)/3)
    Ext30(1, (3*i-2:3*i)) = Ext30(1, (3*i-2:3*i))*Rot;
end

%Helmet Coordinate System (H)
kH_E30 = (Ext30(1,4:6) '-Ext30(1,1:3)')/norm(Ext30(1,4:6) '-Ext30(1,1:3)');
iH_E30 = cross((Ext30(1,7:9) '-Ext30(1,1:3)') , kH_E30)/norm(cross((Ext30(1,7:9) '-
Ext30(1,1:3)') , kH_E30));
jH_E30 = cross(kH_E30, iH_E30)/norm(cross(kH_E30, iH_E30));
r_g_nH_E30 = [iH_E30 jH_E30 kH_E30];

r_H_nH_E30 = r_H_g_FE*r_g_nH_E30;
r_nH_H_E30 = r_H_nH_E30';

F_E30{N} = r_C_H_FE*r_H_nH_E30'*r_C_H_FE'*F_FE;
M_E30{N} = [0, F_E30{N}(3,1), F_E30{N}(2,1); F_E30{N}(3,1), 0, F_E30{N}(1,1); F_E30{N}(
(2,1), F_E30{N}(1,1), 0]*cg_FE;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% LAT MOUNT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Centre of Gravity (CG) Calculations
FxLat = 1.507260961;
FyLat = -20.29118432;
FzLat = 1.417519109;
MxLat = -3.549037305;
MyLat = -0.124070385;
MzLat = -0.445058204;
F_Lat = [FxLat;FyLat;FzLat];

ALatcg = [0,FzLat,FyLat;FzLat,0,FxLat;FyLat,FxLat,0];
M_Lat = [MxLat;MyLat;MzLat];
cg_Lat = ALatcg\M_Lat; %--> cg position in force cube CS in m

%Import data for original positioning of mount
FileNameLat = [info.root 'Moment\Helmet-Cube Distances\helmetcube_lat_mount.xlsx'];
SheetLat = 'helmetcube_lat_mount';
[numLat,~,~] = xlsread(FileNameLat,SheetLat,'c7521:a17521');

%Rotate each marker trajectory according to "Rot"
for i = 1:(size(numLat,2)/3)
    numLat(1, (3*i-2:3*i)) = numLat(1, (3*i-2:3*i))*Rot;
end

%Helmet Coordinate System (H)
kH_Lat = (numLat(1,28:30) '-numLat(1,25:27)')/norm(numLat(1,28:30) '-numLat(1,25:27)');
iH_Lat = cross((numLat(1,31:33) '-numLat(1,25:27)') , kH_Lat)/norm(cross((numLat(1,31:
33) '-numLat(1,25:27)') , kH_Lat));
jH_Lat = cross(kH_Lat, iH_Lat)/norm(cross(kH_Lat, iH_Lat));
r_g_H_Lat = [iH_Lat jH_Lat kH_Lat];
r_H_g_Lat = r_g_H_Lat';
Ori_H_g_Lat = numLat(1,25:27)';

%Force Cube Coordinate System (C)

```

```

iC_Lat = (numLat(1,13:15)'\-numLat(1,1:3)')/norm(numLat(1,13:15)'\-numLat(1,1:3)');
kC_Lat = cross(iC_Lat, (numLat(1,22:24)'\-numLat(1,1:3)'))/norm(cross(iC_Lat, (numLat
(1,22:24)'\-numLat(1,1:3)')));
jC_Lat = cross(kC_Lat,iC_Lat)/norm(cross(kC_Lat,iC_Lat));
r_g_C_Lat = [iC_Lat jC_Lat kC_Lat];
r_C_g_Lat = r_g_C_Lat';
Ori_C_C_Lat = [0.0539206;0.0612142;3.3667];

%Origin of force cube CS in Global
%point intersecting 3 places (2 diagonals of force cube and front surface)
%plane1--> norm=((f2-f4)x(b2-f4)
Norm1Lat = cross((numLat(1,7:9)'\-numLat(1,13:15)'), (numLat(1,10:12)'\-numLat(1,13:
15)'))/norm(cross((numLat(1,7:9)'\-numLat(1,13:15)'), (numLat(1,10:12)'\-numLat(1,13:
15)')));
BLat(1,1) = numLat(1,13:15)*Norm1Lat;
%plane2--> norm=((f1-f3)x(b3-f3)
Norm2Lat = cross((numLat(1,22:24)'\-numLat(1,1:3)'), (numLat(1,4:6)'\-numLat(1,1:3)'))
/norm(cross((numLat(1,22:24)'\-numLat(1,1:3)'), (numLat(1,4:6)'\-numLat(1,1:3)')));
BLat(2,1) = numLat(1,1:3)*Norm2Lat;
%plane3--> norm=kC_Lat
Norm3Lat=kC_Lat;
BLat(3,1) = numLat(1,1:3)*Norm3Lat;
ALat = [Norm1Lat';Norm2Lat';Norm3Lat'];

Ori_C_g_Lat = ALat\BLat+r_g_C_Lat*Ori_C_C_Lat;

%Position and orientation of Force Cube CS in Helmet CS
r_H_C_Lat = r_H_g_Lat*r_g_C_Lat;
r_C_H_Lat = r_H_C_Lat';
t_H_C_Lat = r_H_g_Lat*(Ori_C_g_Lat-Ori_H_g_Lat);

for N = 1:3

    %Force correction for Neutral Lateral flexion *****
    FileNameNLat = [info.root 'UW010\Raw\MVE\mvc' num2str(N) '-neutrallat.CSV'];
    NeutralLat=csvread(FileNameNLat,10511,0);
    NeutralLat=NeutralLat(10,3:end);

    for i = 1:(size(NeutralLat,2)/3)
        NeutralLat(1,(3*i-2:3*i)) = NeutralLat(1,(3*i-2:3*i))*Rot;
    end

    %Helmet Coordinate System (H)
    kH_NL = (NeutralLat(1,4:6)'\-NeutralLat(1,1:3)')/norm(NeutralLat(1,4:6)'\-NeutralLat
(1,1:3)');
    iH_NL = cross((NeutralLat(1,7:9)'\-NeutralLat(1,1:3)'), kH_NL)/norm(cross
((NeutralLat(1,7:9)'\-NeutralLat(1,1:3)'), kH_NL));
    jH_NL = cross(kH_NL,iH_NL)/norm(cross(kH_NL,iH_NL));
    r_g_nH_NL{N} = [iH_NL jH_NL kH_NL];

    r_H_nH_NL = r_H_g_Lat*r_g_nH_NL{N};
    r_nH_H_NL = r_H_nH_NL';

    F_NL{N} = r_C_H_Lat*r_H_nH_NL'*r_C_H_Lat'*F_Lat;
    M_NL{N} = [0,F_NL{N}(3,1),F_NL{N}(2,1);F_NL{N}(3,1),0,F_NL{N}(1,1);F_NL{N}(2,1),

```

```

F_NL{N}(1,1),0]*cg_Lat;

%Force correction for 20lat *****
FileNameLat20 = [info.root 'UW010\Raw\MVE\mvc' num2str(N) '-20lat.CSV'];
Lat20 = csvread(FileNameLat20,10511,0);
Lat20 = Lat20(10,3:end);

for i = 1:(size(Lat20,2)/3)
    Lat20(1,(3*i-2:3*i)) = Lat20(1,(3*i-2:3*i))*Rot;
end

%Helmet Coordinate System (H)
kH_L20 = (Lat20(1,4:6)'\-Lat20(1,1:3)')/norm(Lat20(1,4:6)'\-Lat20(1,1:3)');
iH_L20 = cross((Lat20(1,7:9)'\-Lat20(1,1:3)'),kH_L20)/norm(cross((Lat20(1,7:9)'\-
Lat20(1,1:3)'),kH_L20));
jH_L20 = cross(kH_L20,iH_L20)/norm(cross(kH_L20,iH_L20));
r_g_nH_L20 = [iH_L20 jH_L20 kH_L20];

r_H_nH_L20 = r_H_g_Lat*r_g_nH_L20;
r_nH_H_L20 = r_H_nH_L20';

F_L20{N} = r_C_H_Lat*r_H_nH_L20'*r_C_H_Lat'*F_Lat;
M_L20{N} = [0,F_L20{N}(3,1),F_L20{N}(2,1);F_L20{N}(3,1),0,F_L20{N}(1,1);F_L20{N}(
(2,1),F_L20{N}(1,1),0)*cg_Lat;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ROT MOUNT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

FxRot = 6.070768879;
FyRot = -22.09268923;
FzRot = -0.091236942;
MxRot = -0.681027175;
MyRot = -0.158859211;
MzRot = -3.503072155;
F_Rot = [FxRot;FyRot;FzRot];

ARotcg = [0,FzRot,FyRot;FzRot,0,FxRot;FyRot,FxRot,0];
M_Rot = [MxRot;MyRot;MzRot];
cg_Rot = ARotcg\M_Rot; %--> cg position in force cube CS in m

ARotcg = [0,FzRot,FyRot;FzRot,0,FxRot;FyRot,FxRot,0];
M_Rot = [MxRot;MyRot;MzRot];

cg_Rot = ARotcg\M_Rot; %--> cg position in force cube CS in m

% Import data for original positioning of mount
FileNameRot = [info.root 'Moment\Helmet-Cube Distances\helmetcube_rot_mount.xlsx'];
SheetRot = 'helmetcube_rot_mount';
[numRot,~,~] = xlsread(FileNameRot,SheetRot,'c7521:a17521');

%Rotate each marker trajectory according to "Rot"
for i = 1:(size(numRot,2)/3)
    numRot(1,(3*i-2:3*i)) = numRot(1,(3*i-2:3*i))*Rot;
end

```

%Helmet Coordinate System (H)

```

kH_Rot = (numRot(1,28:30)'\-numRot(1,25:27)')/norm(numRot(1,28:30)'\-numRot(1,25:27)');
iH_Rot = cross((numRot(1,31:33)'\-numRot(1,25:27)'),kH_Rot)/norm(cross((numRot(1,31:33)'\-numRot(1,25:27)'),kH_Rot));
jH_Rot = cross(kH_Rot,iH_Rot)/norm(cross(kH_Rot,iH_Rot));
r_g_H_Rot = [iH_Rot jH_Rot kH_Rot];
r_H_g_Rot = r_g_H_Rot';
Ori_H_g_Rot = numRot(1,25:27)';

```

%Force Cube Coordinate System (C)

```

iC_Rot = (numRot(1,13:15)'\-numRot(1,1:3)')/norm(numRot(1,13:15)'\-numRot(1,1:3)');
kC_Rot = cross(iC_Rot,(numRot(1,22:24)'\-numRot(1,1:3)'))/norm(cross(iC_Rot,(numRot(1,22:24)'\-numRot(1,1:3)')));
jC_Rot = cross(kC_Rot,iC_Rot)/norm(cross(kC_Rot,iC_Rot));
r_g_C_Rot = [iC_Rot jC_Rot kC_Rot];
r_C_g_Rot = r_g_C_Rot';
Ori_C_C_Rot = [0.0539206;0.0612142;3.3667];

```

%Origin of force cube CS in Global

%point intersecting 3 places (2 diagonals of force cube and front surface)

```

%plane1--> norm=((f2-f4)x(b2-f4)
Norm1Rot = cross((numRot(1,7:9)'\-numRot(1,13:15)'),(numRot(1,10:12)'\-numRot(1,13:15)'))/norm(cross((numRot(1,7:9)'\-numRot(1,13:15)'),(numRot(1,10:12)'\-numRot(1,13:15)')));
BRot(1,1) = numRot(1,13:15)*Norm1Rot;
%plane2--> norm=((f1-f3)x(b3-f3)
Norm2Rot = cross((numRot(1,22:24)'\-numRot(1,1:3)'),(numRot(1,4:6)'\-numRot(1,1:3)'))/norm(cross((numRot(1,22:24)'\-numRot(1,1:3)'),(numRot(1,4:6)'\-numRot(1,1:3)')));
BRot(2,1) = numRot(1,1:3)*Norm2Rot;
%plane3--> norm=kC_Rot
Norm3Rot = kC_Rot;
BRot(3,1) = numRot(1,1:3)*Norm3Rot;
ARot = [Norm1Rot';Norm2Rot';Norm3Rot'];

```

```

Ori_C_g_Rot = ARot\BRot+r_g_C_Rot*Ori_C_C_Rot;

```

% Position and orientation of Force cube CS according to Helmet CS

```

r_H_C_Rot = r_H_g_Rot*r_g_C_Rot;
r_C_H_Rot = r_H_C_Rot';
t_H_C_Rot = r_H_g_Rot*(Ori_C_g_Rot-Ori_H_g_Rot);

```

```

for N = 1:3

```

% Force correction for Neutral Rotation *****

```

FileNameNRot = [info.root 'UW010\Raw\MVE\mvc' num2str(N) '-neutralrot.CSV'];
NeutralRot = csvread(FileNameNRot,10511,0);
NeutralRot = NeutralRot(10,3:end);

```

```

for i = 1:(size(NeutralRot,2)/3)
    NeutralRot(1,(3*i-2:3*i)) = NeutralRot(1,(3*i-2:3*i))*Rot;
end

```

%Helmet Coordinate System (H)

```

kH_NR = (NeutralRot(1,4:6)'\-NeutralRot(1,1:3)')/norm(NeutralRot(1,4:6)'\-NeutralRot(1,1:3)');

```

```

(1,1:3)');
    iH_NR = cross((NeutralRot(1,7:9)'\-NeutralRot(1,1:3)'),kH_NR)/norm(cross
((NeutralRot(1,7:9)'\-NeutralRot(1,1:3)'),kH_NR));
    jH_NR = cross(kH_NR,iH_NR)/norm(cross(kH_NR,iH_NR));
    r_g_nH_NR{N} = [iH_NR jH_NR kH_NR];

    r_H_nH_NR = r_H_g_Rot*r_g_nH_NR{N};
    r_nH_H_NR = r_H_nH_NR';

    F_NR{N} = r_C_H_Rot*r_H_nH_NR'*r_C_H_Rot'*F_Rot;
    M_NR{N} = [0,F_NR{N}(3,1),F_NR{N}(2,1);F_NR{N}(3,1),0,F_NR{N}(1,1);F_NR{N}(2,1),
F_NR{N}(1,1),0]*cg_Rot;

    %Force correction for 45rot *****
    FileNameRot45 = [info.root 'UW010\Raw\MVE\mvc' num2str(N) '-45rot.CSV'];
    Rot45 = csvread(FileNameRot45,10511,0);
    Rot45 = Rot45(10,3:end);

    for i = 1:(size(Rot45,2)/3)
        Rot45(1,(3*i-2:3*i)) = Rot45(1,(3*i-2:3*i))*Rot;
    end

    %Helmet Coordinate System (H)
    kH_R45 = (Rot45(1,4:6)'\-Rot45(1,1:3)')/norm(Rot45(1,4:6)'\-Rot45(1,1:3)');
    iH_R45 = cross((Rot45(1,7:9)'\-Rot45(1,1:3)'),kH_R45)/norm(cross((Rot45(1,7:9)'\-
Rot45(1,1:3)'),kH_R45));
    jH_R45 = cross(kH_R45,iH_R45)/norm(cross(kH_R45,iH_R45));
    r_g_nH_R45 = [iH_R45 jH_R45 kH_R45];

    r_H_nH_R45 = r_H_g_Rot*r_g_nH_R45;
    r_nH_H_R45 = r_H_nH_R45';

    F_R45{N} = r_C_H_Rot*r_H_nH_R45'*r_C_H_Rot'*F_Rot;
    M_R45{N} = [0,F_R45{N}(3,1),F_R45{N}(2,1);F_R45{N}(3,1),0,F_R45{N}(1,1);F_R45{N}
(2,1),F_R45{N}(1,1),0]*cg_Rot;

    %Force correction for 45rot45flex *****
    FileNameRot45Flex45 = [info.root 'UW010\Raw\MVE\mvc' num2str(N) '-45rot45flex.
CSV'];
    Rot45Flex45 = csvread(FileNameRot45Flex45,10511,0);
    Rot45Flex45 = Rot45Flex45(10,3:end);

    for i = 1:(size(Rot45Flex45,2)/3)
        Rot45Flex45(1,(3*i-2:3*i)) = Rot45Flex45(1,(3*i-2:3*i))*Rot;
    end

    %Helmet Coordinate System (H)
    kH_R45F45 = (Rot45Flex45(1,4:6)'\-Rot45Flex45(1,1:3)')/norm(Rot45Flex45(1,4:6)'\-
Rot45Flex45(1,1:3)');
    iH_R45F45 = cross((Rot45Flex45(1,7:9)'\-Rot45Flex45(1,1:3)'),kH_R45F45)/norm(cross
((Rot45Flex45(1,7:9)'\-Rot45Flex45(1,1:3)'),kH_R45F45));
    jH_R45F45 = cross(kH_R45F45,iH_R45F45)/norm(cross(kH_R45F45,iH_R45F45));
    r_g_nH_R45F45 = [iH_R45F45 jH_R45F45 kH_R45F45];

    r_H_nH_R45F45 = r_H_g_Rot*r_g_nH_R45F45;

```

```

r_nH_H_R45F45 = r_H_nH_R45F45';

F_R45F45{N} = r_C_H_Rot*r_H_nH_R45F45'*r_C_H_Rot'*F_Rot;
M_R45F45{N} = [0,F_R45F45{N}(3,1),F_R45F45{N}(2,1);F_R45F45{N}(3,1),0,F_R45F45{N}(1,1);F_R45F45{N}(2,1),F_R45F45{N}(1,1),0]*cg_Rot;

%Force correction for 45rot30ext *****
FileNameRot45Ext30 = [info.root 'UW010\Raw\MVE\mvc' num2str(N) '-45rot30ext.CSV'];
Rot45Ext30 = csvread(FileNameRot45Ext30,10511,0);
Rot45Ext30 = Rot45Ext30(10,3:end);

for i = 1:(size(Rot45Ext30,2)/3)
    Rot45Ext30(1,(3*i-2:3*i)) = Rot45Ext30(1,(3*i-2:3*i))*Rot;
end

%Helmet Coordinate System (H)
kH_R45E30 = (Rot45Ext30(1,4:6) '-Rot45Ext30(1,1:3)')/norm(Rot45Ext30(1,4:6) '-Rot45Ext30(1,1:3)');
iH_R45E30 = cross((Rot45Ext30(1,7:9) '-Rot45Ext30(1,1:3)'),kH_R45E30)/norm(cross((Rot45Ext30(1,7:9) '-Rot45Ext30(1,1:3)'),kH_R45E30));
jH_R45E30 = cross(kH_R45E30,iH_R45E30)/norm(cross(kH_R45E30,iH_R45E30));
r_g_nH_R45E30 = [iH_R45E30 jH_R45E30 kH_R45E30];

r_H_nH_R45E30 = r_H_g_Rot*r_g_nH_R45E30;
r_nH_H_R45E30 = r_H_nH_R45E30';

F_R45E30{N} = r_C_H_Rot*r_H_nH_R45E30'*r_C_H_Rot'*F_Rot;
M_R45E30{N} = [0,F_R45E30{N}(3,1),F_R45E30{N}(2,1);F_R45E30{N}(3,1),0,F_R45E30{N}(1,1);F_R45E30{N}(2,1),F_R45E30{N}(1,1),0]*cg_Rot;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Export results %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for N = 1:3
    Headers = {'Conditions','Fx','Fy','Fz','Mx','My','Mz'};
    Results(1,:) = horzcat('neutralflex',num2cell(F_NF{N}'),num2cell(M_NF{N}'));
    Results(2,:) = horzcat('neutralext',num2cell(F_NE{N}'),num2cell(M_NE{N}'));
    Results(3,:) = horzcat('45flex',num2cell(F_F45{N}'),num2cell(M_F45{N}'));
    Results(4,:) = horzcat('30ext',num2cell(F_E30{N}'),num2cell(M_E30{N}'));
    Results(5,:) = horzcat('neutrallat',num2cell(F_NL{N}'),num2cell(M_NL{N}'));
    Results(6,:) = horzcat('20lat',num2cell(F_L20{N}'),num2cell(M_L20{N}'));
    Results(7,:) = horzcat('neutralrot',num2cell(F_NR{N}'),num2cell(M_NR{N}'));
    Results(8,:) = horzcat('45rot',num2cell(F_R45{N}'),num2cell(M_R45{N}'));
    Results(9,:) = horzcat('45rot45flex',num2cell(F_R45F45{N}'),num2cell(M_R45F45{N}'));
    Results(10,:) = horzcat('45rot30ext',num2cell(F_R45E30{N}'),num2cell(M_R45E30{N}'));

    TotalResults = [Headers;Results];
    sheets = {'MVC1','MVC2','MVC3'};
    xlswrite([info.root 'Moment\Correction Factors\CorrectionFactors_MVE.xlsx'],TotalResults,sheets{N});
end

%Collect helemt_cube Rotations and Translations

```

```
%(used in MVC_conversion_to_neck)
H_C.r_H_C_FE = r_H_C_FE;
H_C.r_C_H_FE = r_C_H_FE;
H_C.t_H_C_FE = t_H_C_FE;
H_C.r_H_C_Lat = r_H_C_Lat;
H_C.r_C_H_Lat = r_C_H_Lat;
H_C.t_H_C_Lat = t_H_C_Lat;
H_C.r_H_C_Rot = r_H_C_Rot;
H_C.r_C_H_Rot = r_C_H_Rot;
H_C.t_H_C_Rot = t_H_C_Rot;

H_g.r_H_g_FE = r_H_g_FE;
H_g.r_H_g_Lat = r_H_g_Lat;
H_g.r_H_g_Rot = r_H_g_Rot;
H_g.r_g_nH_NE = (r_g_nH_NE{1} + r_g_nH_NE{2} + r_g_nH_NE{3})/3;
H_g.r_g_nH_NF = (r_g_nH_NF{1} + r_g_nH_NF{2} + r_g_nH_NF{3})/3;
H_g.r_g_nH_NL = (r_g_nH_NL{1} + r_g_nH_NL{2} + r_g_nH_NL{3})/3;
H_g.r_g_nH_NR = (r_g_nH_NR{1} + r_g_nH_NR{2} + r_g_nH_NR{3})/3;
F_Corr.F_FE = F_FE;
F_Corr.F_Lat = F_Lat;
F_Corr.F_Rot = F_Rot;

save([info.root 'Moment\Transformations\RotationTranslationMatrix.
mat'], 'H_C', 'H_g', 'F_Corr');
```

```

function [nFEout,nLATout,nROTout] = load_neutral(info,subid,Rot)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Loads a neutral trial from each mount type.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

headerRow = 10511;
headerCol = 2;

% moment_filter_posture(info,indata,Rot)

%FE
indata = importdata([info.root subid '\Raw\MVE\mvc1-neutralflex.CSV'],'',headerRow);
import
nFE1 = indata.data(:,headerCol+1:end);

for l = 1:size(nFE1,1)
    for m = 1:(size(nFE1,2)/3)
        nFE(l,(3*m-2:3*m)) = nFE1(l,(3*m-2:3*m))*Rot;
    end
end

%LAT
indata = importdata([info.root subid '\Raw\MVE\mvc1-neutrallat.CSV'],'',headerRow);
import
nLAT1 = indata.data(:,headerCol+1:end);

for l = 1:size(nLAT1,1)
    for m = 1:(size(nLAT1,2)/3)
        nLAT(l,(3*m-2:3*m)) = nLAT1(l,(3*m-2:3*m))*Rot;
    end
end

%ROT
indata = importdata([info.root subid '\Raw\MVE\mvc1-neutralrot.CSV'],'',headerRow);
import
nROT1 = indata.data(:,headerCol+1:end);

for l = 1:size(nROT1,1)
    for m = 1:(size(nROT1,2)/3)
        nROT(l,(3*m-2:3*m)) = nROT1(l,(3*m-2:3*m))*Rot;
    end
end

nFEout = nFE(200,:);
nLATout = nLAT(200,:);
nROTout = nROT(200,:);

```



```
function force_filt = moment_filter_force(info,forcedata,shunt_key,shuntrow)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Filters force and moment recordings.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Convert force data from volts (V) to Newtons (N)
for k = 1:size(forcedata,1)
    force_N(k,1:6) = ((forcedata(k,1:6)-cell2mat(shunt_key(shuntrow,26:31))).*cell2mat(
(shunt_key(shuntrow,2:7)))./cell2mat(shunt_key(shuntrow,20:25)));
end

Wn = info.fc_force/(info.fs_force/2);

%Filter Force and Moment Data
[b,a] = butter(2,Wn,'low');
force_filt = filtfilt(b,a,force_N);
```

```
function outdata = moment_filter_posture(info,indata,Rot)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, Caroline Joseph, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Filters marker position data, and rotates all markers according
%to Rot.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Rotate Posture Data
for l = 1:size(indata,1)
    for m = 1:(size(indata,2)/3)
        data_rot(l,(3*m-2:3*m)) = indata(l,(3*m-2:3*m))*Rot;
    end
end

%Pad and Filter Posture Data
for l = 1:350
    if l < 350
        for m = 1:30
            pad_data(((30*l)+m-30), :) = data_rot(l, :)+(m/30)*(data_rot((l+1), :)-
data_rot(l, :)));
        end
    elseif l == 350
        for m = 1:30
            pad_data(((30*l)+m-30), :) = data_rot(l, :)+(m/30)*(data_rot((l), :)-
data_rot(l-1, :)));
        end
    end
end

Wn = info.fc_force/(info.fs_force/2);

[b,a] = butter(2,Wn,'low');
outdata = filtfilt(b,a,pad_data);
```

```

function [neck_moment,neck_angles] = moment_convert_to_neckCS(filename,info,subid,
forcedata,posturedata,trialtype,nFE,nLAT,nROT,H_C,H_g,F_Corr,outData)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, Caroline Joseph, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Transforms neck force and moment values into the neck local
%system, and extracts peak values from each trial.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

conditions =
{'neutralflex','neutralext','45flex','30ext','neutrallat','20lat','neutralrot','45rot',
'45rot30ext','45rot45flex'};

%FE Movements *****
if strcmp(trialtype,conditions(1))||strcmp(trialtype,conditions(2))||strcmp(trialtype,
conditions(3))||strcmp(trialtype,conditions(4))
    r_H_C = H_C.r_H_C_FE;
    r_C_H = H_C.r_C_H_FE;
    t_H_C = H_C.t_H_C_FE;
    r_H_g = H_g.r_H_g_FE;
    F_Cor = F_Corr.F_FE;

    %Identify appropriate neutral posture file based on trial posture
    neutraldata = nFE;
end

%LAT Movements *****
if strcmp(trialtype,conditions(5))||strcmp(trialtype,conditions(6))
    r_H_C = H_C.r_H_C_Lat;
    r_C_H = H_C.r_C_H_Lat;
    t_H_C = H_C.t_H_C_Lat;
    r_H_g = H_g.r_H_g_Lat;
    F_Cor = F_Corr.F_Lat;

    %Identify appropriate neutral posture file based on trial posture
    neutraldata = nLAT;
end

%ROT Movements *****
if strcmp(trialtype,conditions(7))||strcmp(trialtype,conditions(8))||strcmp(trialtype,
conditions(9))||strcmp(trialtype,conditions(10))
    r_H_C = H_C.r_H_C_Rot;
    r_C_H = H_C.r_C_H_Rot;
    t_H_C = H_C.t_H_C_Rot;
    r_H_g = H_g.r_H_g_Rot;
    F_Cor = F_Corr.F_Rot;

    %Identify appropriate neutral posture file based on trial posture
    neutraldata = nROT;
end

%Calculate neutral posture helmet coordinate system (global and local)
kH_N = (neutraldata(4:6)-neutraldata(1:3))/norm(neutraldata(4:6)-neutraldata(1:3));

```

```

iH_N = cross((neutraldata(7:9)-neutraldata(1:3)),kH_N)/norm(cross((neutraldata(7:9)-
neutraldata(1:3)),kH_N));
jH_N = cross(kH_N,iH_N)/norm(cross(kH_N,iH_N));
r_g_HN = [iH_N' jH_N' kH_N'];
r_H_gN = r_g_HN';

for k = 1:size(posturedata,1)
    %Trial Helmet CS
    kH_Data = (posturedata(k,4:6)-posturedata(k,1:3))/norm(posturedata(k,4:6)-
posturedata(k,1:3));
    iH_Data = cross((posturedata(k,7:9)-posturedata(k,1:3)),kH_Data)/norm(cross
((posturedata(k,7:9)-posturedata(k,1:3)),kH_Data));
    jH_Data = cross(kH_Data,iH_Data)/norm(cross(kH_Data,iH_Data));
    r_g_HData = [iH_Data' jH_Data' kH_Data'];
    r_H_gData = r_g_HData';
    Ori_HData_g = posturedata(k,1:3)';

    %Trial Neck CS --> oriented with torso
    jNe_Data_tempo = (((posturedata(k,22:24)+posturedata(k,19:21))/2)-((posturedata(k,
22:24)+posturedata(k,25:27))/2));
    jNe_Data = jNe_Data_tempo/norm(jNe_Data_tempo);
    kNe_Data = cross((posturedata(k,19:21)-posturedata(k,22:24)),jNe_Data)/norm(cross
((posturedata(k,19:21)-posturedata(k,22:24)),jNe_Data));
    iNe_Data = cross(jNe_Data,kNe_Data)/norm(cross(jNe_Data,kNe_Data));
    r_g_NeData = [iNe_Data' jNe_Data' kNe_Data'];
    r_Ne_gData = r_g_NeData';
    Ori_NeData_g = (posturedata(k,19:21)'+posturedata(k,22:24)')/2; % 50% depth based
McGill et al 1988

    %Corrected forces at force cube
    r_H_nH = r_H_g*r_g_HData;
    F_Offset = r_C_H*r_H_nH'*r_H_C*F_Cor;

    forcedata(k,1:3) = forcedata(k,1:3)-F_Offset';

    %Change orientation of forces based on neck coordinate system
    r_H_NeData = r_H_gData*r_g_NeData;
    r_Ne_HData = r_H_NeData';
    r_C_NeData = r_C_H*r_H_NeData;
    r_Ne_CData = r_C_NeData';
    F_Ne(k,1:3) = (r_Ne_CData*forcedata(k,1:3)')';

    %           %tempo correction that only applies for UW009
    %           F_Ne(k,1:3) = F_Ne(k,1:3)-[25,30,0];

    %Get distances of force Cube compared to Neck coordinate system origin
    t_H_C_Ne = r_Ne_HData*t_H_C;
    t_H_Ne_g = Ori_NeData_g-Ori_HData_g;
    t_H_Ne_Ne = r_Ne_gData*t_H_Ne_g;
    t_C_Ne_Ne = (-t_H_Ne_Ne+t_H_C_Ne)/1000; %to convert to metres

    %Calculate Moments at neck (about origin of CS)
    M_Ne(k,1) = -F_Ne(k,2)*t_C_Ne_Ne(3,1)+F_Ne(k,3)*t_C_Ne_Ne(2,1);
    M_Ne(k,2) = F_Ne(k,1)*t_C_Ne_Ne(3,1)-F_Ne(k,3)*t_C_Ne_Ne(1,1);
    M_Ne(k,3) = -F_Ne(k,1)*t_C_Ne_Ne(2,1)+F_Ne(k,2)*t_C_Ne_Ne(1,1);

```

```

t_C_Ne_NeTotal(k,1:3) = t_C_Ne_Ne';

%Head orientation between neutral and current posture
r_H_HData = r_H_gN*r_g_HData;

%FE ZYX
if strcmp(trialtype,conditions(1))||strcmp(trialtype,conditions(4))||strcmp(
(trialtype,conditions(2))||strcmp(trialtype,conditions(3)) %FE
    RotAngle = asind(-r_H_HData(3,1));
    FEAngle = atan2(r_H_HData(2,1),r_H_HData(1,1))*180/pi;
    LatAngle = atan2(r_H_HData(3,2),r_H_HData(3,3))*180/pi;
end

%Lat XYZ
if strcmp(trialtype,conditions(5))||strcmp(trialtype,conditions(6)) % Lat
    RotAngle = asind(r_H_HData(1,3));
    LatAngle = atan2(-r_H_HData(2,3),r_H_HData(3,3))*180/pi;
    FEAngle = atan2(-r_H_HData(1,2),r_H_HData(1,1))*180/pi;
end

%Rot YZX
if strcmp(trialtype,conditions(7))||strcmp(trialtype,conditions(8))||strcmp(
(trialtype,conditions(9))||strcmp(trialtype,conditions(10)) % Rot
    FEAngle = asind(r_H_HData(2,1));
    RotAngle = atan2(-r_H_HData(3,1),r_H_HData(1,1))*180/pi;
    LatAngle = atan2(-r_H_HData(2,3),r_H_HData(2,2))*180/pi;
end

% Angles --> x,y,z
Angles(k,1) = LatAngle;
Angles(k,2) = RotAngle;
Angles(k,3) = FEAngle;
end

%Calculate Peak Force *****
[~,Fmaxloc] = max(abs(F_Ne));
F_max = [F_Ne(Fmaxloc(1),1) F_Ne(Fmaxloc(2),2) F_Ne(Fmaxloc(3),3)];

%Calculate Peak Moment *****
%OVERALL PEAK
[~,Mmaxloc] = max(abs(M_Ne));
M_max = [M_Ne(Mmaxloc(1),1) M_Ne(Mmaxloc(2),2) M_Ne(Mmaxloc(3),3)];

%Define Output Variables
neck_moment = [F_max M_max];
neck_angles = mean(Angles);

%Export Data *****
if outData == 1
    %Force and Moment Values in Neck Coordinate System
    dlmwrite([info.root subid '\Force-Moment Local (Neck)\' filename(1:end-4)
'_forceL.csv'],[F_Ne M_Ne]);
end

```



```
function track_progress(N)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Provides output to track process of data processing.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Track Processing Progress *****
if N == 10 || N == 20 || N == 30 || N == 40 || N == 50 || N == 60 || N == 70 || N == 80 || N == 90 || N == 100 || N == 110 || N == 120
    fprintf([num2str(N) '\n'])
else
    fprintf('.')
end
```

Appendix A3: MATLAB code for processing Kinematic (Motion) data.

[illegible]


```
(startfile:numfiles)') cell2mat(posture(startfile:numfiles)') cell2mat(out_rom
(startfile:numfiles)')],info.subjects{M},'A3');
    end

    fprintf([info.subjects{M} 'Complete.\n'])
    toc(tstart)
end
```

```
function [pathname,filename,numfiles] = file_loader(info,index)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Finds appropriate file path and filenames based on specified
%root folder and trial type variables. Returns "pathname", "filename",
%and the number of files contained in "filename".
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%SELECT FILES TO PROCESS
%Extract list of filenames for subject MVE folder
pathname = [info.root info.subjects{index} '\Raw\' info.filetype '\'];
filenames = dir(pathname(1:end-1));
filename = {filenames.name};
filename(1:2) = [];

%Determine number of active files (# of files pulled from MVE directory)
numfiles = size(filename,2);
```

```

function [l5loc,DC] = motion_L5reconstruct(pathname)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION:
%Reads in motion (MOCAP) data files for a static standing trial, and
%calculates the mean translation vector from all thorax markers to the L5
%marker (used for L5 marker reconstruction in experimental trials). Reads a
%experimental neutral posture file, and constructs direction cosine (DC)
%matrices for the head and thorax (used to calculate neck angles relative
%to neutral).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Import File Data *****
headerRow = {7511,7511,30011,30011}; %Number of header rows in data files (i.e. the
EMG data)
headerCol = 2; %Number of header columns in data files
tag = {'standing_hat.CSV','standing.CSV','neutral_hat.CSV','neutral.CSV'}; %Static
trial filenames

for H = 1:4
    imdata = importdata([pathname(1:(end-4)) tag{H}],',',headerRow{H});
    markers = imdata.data(:,headerCol+1:end); %Removes 2 header columns

    %DEFINE HEAD MARKERS *****
    switch H
        case{1,3} %no helmet
            headtype = 0;
            m.hat1 = mean(markers(:,1:3));
            m.hat2 = mean(markers(:,4:6));
            m.hat3 = mean(markers(:,7:9));
            m.hat4 = mean(markers(:,10:12));
            m.hat5 = mean(markers(:,13:15));
            m.rac = mean(markers(:,16:18));
            m.lac = mean(markers(:,19:21));
            m.ss = mean(markers(:,22:24));
            m.c7 = mean(markers(:,25:27));
            m.xp = mean(markers(:,28:30));
            if H == 1
                m.l5 = mean(markers(:,31:33));
            end
        case{2,4} %helmet
            headtype = 1;
            m.HL = mean(markers(:,1:3));
            m.HR = mean(markers(:,4:6));
            m.HT = mean(markers(:,7:9));
            m.rac = mean(markers(:,13:15));
            m.lac = mean(markers(:,16:18));
            m.ss = mean(markers(:,19:21));
            m.c7 = mean(markers(:,22:24));
            m.xp = mean(markers(:,25:27));
            if H == 2
                m.l5 = mean(markers(:,28:30));
            end
    end
end

```

```

end
end

switch H
case{1,2}
%CREATE L5 LOCATION VECTORS *****
%Develop Thorax Local System (Temporary)
lacrac = m.rac-m.lac;
zaxis1 = lacrac/norm(lacrac);
midtorso = (m.lac+m.rac)/2;
temp = (m.ss-midtorso)/norm(m.ss-midtorso);
yaxis1 = cross(zaxis1,temp)/norm(cross(zaxis1,temp));
xaxis1 = cross(yaxis1,zaxis1);
rot_matrix = [xaxis1; yaxis1; zaxis1];

lac_15 = m.l5-m.lac;
rac_15 = m.l5-m.rac;
ss_15 = m.l5-m.ss;
c7_15 = m.l5-m.c7;
xp_15 = m.l5-m.xp;

%L5 Location Vectors
l5loc{H}.lac = (rot_matrix*lac_15)';
l5loc{H}.rac = (rot_matrix*rac_15)';
l5loc{H}.ss = (rot_matrix*ss_15)';
l5loc{H}.c7 = (rot_matrix*c7_15)';
l5loc{H}.xp = (rot_matrix*xp_15)';

case{3,4}
%Develop Thorax Local System (Temporary) to Reconstruct L5
lacrac = m.rac-m.lac;
zaxis1 = lacrac/norm(lacrac);
midtorso = (m.lac+m.rac)/2;
temp = (m.ss-midtorso)/norm(m.ss-midtorso);
yaxis1 = cross(zaxis1,temp)/norm(cross(zaxis1,temp));
xaxis1 = cross(yaxis1,zaxis1);
rot_matrix = [xaxis1; yaxis1; zaxis1];

%Reconstruct from each thorax marker
l5.lac = (m.lac' + rot_matrix'*l5loc{H-2}.lac)';
l5.rac = (m.rac' + rot_matrix'*l5loc{H-2}.rac)';
l5.ss = (m.ss' + rot_matrix'*l5loc{H-2}.ss)';
l5.c7 = (m.c7' + rot_matrix'*l5loc{H-2}.c7)';
l5.xp = (m.xp' + rot_matrix'*l5loc{H-2}.xp)';

%Reconstructed L5 Location
m.l5 = mean([l5.lac; l5.rac; l5.ss; l5.c7; l5.xp]);

%CALCULATE HEAD AND THORAX DIRECTION COSINE MATRICES *****
%To normalize neck angle to a neutral posture
[i.head,j.head,k.head,o.head] = calculate_lcs(m,headtype);
[i.thor,j.thor,k.thor,o.thor] = calculate_lcs(m,2);

DC{H-2}.head = [i.head' j.head' k.head']; %3 x 3
DC{H-2}.thor = [i.thor' j.thor' k.thor']; %3 x 3

```

```
end  
clear m  
end
```

```

function [iaxis,jaxis,kaxis,origin] = calculate_lcs(m,segment)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Calculates a segment local coordinate system (i,j,k) for the
%head (with or without helmet) or thorax.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if segment == 0
    %Calculate Head Local Coordinate System (NO HELMET)
    kaxis = bsxfun(@rdivide,(m.hat1-m.hat2),normr(m.hat1-m.hat2)); %left-right axis, ✓
    +k is right
    itemp = bsxfun(@rdivide,(m.hat4-m.hat2),normr(m.hat4-m.hat2));
    jaxis = bsxfun(@rdivide,cross(kaxis,itemp),normr(cross(kaxis,itemp))); %j is up ✓
    (cranial)
    iaxis = bsxfun(@rdivide,cross(jaxis,kaxis),normr(cross(jaxis,kaxis))); %i is ✓
    forward
    origin = m.c7;
elseif segment == 1
    %Calculate Head Local Coordinate System (WITH HELMET)
    midhead = (m.HR+m.HL)/2;
    jaxis = bsxfun(@rdivide,(m.HT-midhead),normr(m.HT-midhead)); %j is up (cranial)
    khead_temp = bsxfun(@rdivide,(m.HR-midhead),normr(m.HR-midhead));
    iaxis = bsxfun(@rdivide,cross(jaxis,khead_temp),normr(cross(jaxis,khead_temp))); % ✓
    +i is forward
    kaxis = bsxfun(@rdivide,cross(iaxis,jaxis),normr(cross(iaxis,jaxis))); %left-right ✓
    axis, +k is right
    origin = m.c7;
elseif segment == 2
    %Calculate Thorax Local Coordinate System
    MTS = (m.c7+m.ss)/2; %mid torso superior
    MTI = (m.l5+m.xp)/2; %mid torso inferior
    jaxis = bsxfun(@rdivide,(MTS-MTI),normr(MTS-MTI)); %long axis, +k is up
    SS_C7 = m.c7-m.ss;
    SS_XP = m.xp-m.ss;
    kthor_temp = bsxfun(@rdivide,cross(SS_C7,SS_XP),normr(cross(SS_C7,SS_XP)));
    iaxis = bsxfun(@rdivide,cross(jaxis,kthor_temp),normr(cross(jaxis,kthor_temp))); % ✓
    +i is forward
    kaxis = bsxfun(@rdivide,cross(iaxis,jaxis),normr(cross(iaxis,jaxis))); %k is ✓
    right
    origin = m.c7;
end

```



```

function [t_num,h_num,p_num,nohelm] = file_characteristics(filename)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Break down filename into trial characteristics.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Break down filename into trial characteristics
dashloc = strfind(filename,'-');
t_str = filename(1:dashloc(1)-1); %TRIAL NUMBER ID
h_str = filename(dashloc(1)+1:dashloc(2)-1); %HELMET TYPE ID
p_str = filename(dashloc(2)+1:end-4); %POSTURE ID

%Assign numerical code to each characteristics (makes ofr easier file I/O)
%TRIAL NUMBER ID
t_num = str2num(t_str);
nohelm = 2; %designates as a helmet (not "nohelmet")

%HELMET TYPE ID
switch h_str
    case{'nohelmet'}
        h_num = 1;
        nohelm = 1;
    case{'hOnly'}
        h_num = 2;
    case{'hNVGup'}
        h_num = 3;
    case{'hNVGdown'}
        h_num = 4;
    case{'hCWup'}
        h_num = 5;
    case{'hCWdown'}
        h_num = 6;
end

%POSTURE ID
switch p_str
    case{'neutral'}
        p_num = 1;
    case{'45flex'}
        p_num = 2;
    case{'30ext'}
        p_num = 3;
    case{'20lat'}
        p_num = 4;
    case{'45rot'}
        p_num = 5;
    case{'45rot45flex'}
        p_num = 6;
    case{'45rot30ext'}
        p_num = 7;
end

```

```

function [X,Y,Z] = motion_head_angle(sample_f,markers,l5loc,DC_N,helmet,filename,
plotData)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION:
%1) Imports a set of L5 location vectors (l5loc) and simulates L5
%marker location based on relation ship to all other thorax markers.
%2) Calculates neck Euler angles of rotation relative to a neutral standing
%posture.
%3) Differentiates neck angles to angular velocity (used to define static
%hold phase of motion trials).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%DEFINE HEAD MARKERS *****
switch helmet
    case{1} %nohelmet
        m.hat1 = markers(:,1:3);
        m.hat2 = markers(:,4:6);
        m.hat3 = markers(:,7:9);
        m.hat4 = markers(:,10:12);
        m.hat5 = markers(:,13:15);
        m.rac = markers(:,16:18);
        m.lac = markers(:,19:21);
        m.ss = markers(:,22:24);
        m.c7 = markers(:,25:27);
        m.xp = markers(:,28:30);
        headtype = 0;
    otherwise
        m.HL = markers(:,1:3);
        m.HR = markers(:,4:6);
        m.HT = markers(:,7:9);
        m.rac = markers(:,13:15);
        m.lac = markers(:,16:18);
        m.ss = markers(:,19:21);
        m.c7 = markers(:,22:24);
        m.xp = markers(:,25:27);
        headtype = 1;
end

samples = length(markers);

%RECONSTRUCT L5 MARKER LOCATION *****
for N = 1:samples
    %Develop Thorax Local System (Temporary) to Reconstruct L5
    lacrac = m.rac(N,:)-m.lac(N,:);
    zaxis1 = lacrac/norm(lacrac);
    midtorso = (m.lac(N,:)+m.rac(N,:))/2;
    temp = (m.ss(N,:)-midtorso)/norm(m.ss(N,:)-midtorso);
    yaxis1 = cross(zaxis1,temp)/norm(cross(zaxis1,temp));
    xaxis1 = cross(yaxis1,zaxis1);
    rot_matrix = [xaxis1; yaxis1; zaxis1];
end

```

```

%Reconstruct from each thorax marker
15.lac(N,:) = (m.lac(N,:) + rot_matrix'*l5loc.lac)';
15.rac(N,:) = (m.rac(N,:) + rot_matrix'*l5loc.rac)';
15.ss(N,:) = (m.ss(N,:) + rot_matrix'*l5loc.ss)';

%Reconstructed L5 Location
m.l5(N,:) = mean([15.lac(N,:); 15.rac(N,:); 15.ss(N,:)]);
end

%CALCULATE HEAD LOCAL SYSTEM *****
[i.head,j.head,k.head,o.head] = calculate_lcs(m,headtype);

%CALCULATE THORAX LOCAL SYSTEM *****
[i.thor,j.thor,k.thor,o.thor] = calculate_lcs(m,2);

%BUILD DIRECTION COSINE (DC) MATRICES *****

for Q = 1:samples
    DC.head(:,:,Q) = [i.head(Q,:) j.head(Q,:) k.head(Q,:)]; %3 x 3 x samples
    DC.thor(:,:,Q) = [i.thor(Q,:) j.thor(Q,:) k.thor(Q,:)]; %3 x 3 x samples

    %Remove "neutral posture bias"
    DC2.head(:,:,Q) = DC_N.head'*DC.head(:,:,Q);
    DC2.thor(:,:,Q) = DC_N.thor'*DC.thor(:,:,Q);

    %Rotation Matrix: Head Relative to Thorax
    DCR = DC_N.thor'*DC_N.head; %3 x 3 x samples
    R.thor_head(:,:,Q) = DC.thor(:,:,Q)'*DC.head(:,:,Q); %3 x 3 x samples

    %Remove "Neutral Posture Bias"
    Ycorr = -asin(DCR(1,3)); %Axial Rotation [+ve is right]
    Xcorr = -atan(DCR(2,3)/DCR(3,3)); %Lateral Bending [+ve is right]
    Zcorr = atan(DCR(1,2)/DCR(1,1)); %Flexion-Extension [+ve is flexion]

    %DECOMPOSE DC MATRIX INTO ROTATION ANGLES *****
    %[XYZ rotation sequence - Wu et al., 2002]
    Y.theta(Q,1) = -asin(R.thor_head(1,3,Q)); %Axial Rotation [+ve is right]
    X.theta(Q,1) = -atan(R.thor_head(2,3,Q)/R.thor_head(3,3,Q)); %Lateral Bending [+ve is right]
    Z.theta1(Q,1) = atan(R.thor_head(1,2,Q)/R.thor_head(1,1,Q)); %Flexion-Extension [+ve is flexion]

    %Remove "Neutral Posture Bias"
    Y.theta(Q,1) = Y.theta(Q,1) - Ycorr;
    X.theta(Q,1) = X.theta(Q,1) - Xcorr;
    Z.theta(Q,1) = Z.theta1(Q,1) - Zcorr;
end

%DIFFERENTIATE ANGLES (into angular velocities) *****
thetaPadX = data_pad(1,X.theta,sample_f,2); %PAD DATA
thetaPadY = data_pad(1,Y.theta,sample_f,2);
thetaPadZ = data_pad(1,Z.theta,sample_f,2);
tempX = (thetaPadX(3:end,:)-thetaPadX(1:end-2,:))/(2*(1/sample_f)); %DIFFERENTIATE DATA

```

```
tempY = (thetaPadY(3:end,:)-thetaPadY(1:end-2,:))/(2*(1/sample_f));
tempZ = (thetaPadZ(3:end,:)-thetaPadZ(1:end-2,:))/(2*(1/sample_f));

%Smooth data for consistent threshold extraction
[b,a] = butter(2,1/(sample_f/2),'low');
tempX2 = filtfilt(b,a,tempX); %FILTER DATA
tempY2 = filtfilt(b,a,tempY);
tempZ2 = filtfilt(b,a,tempZ);
X.omega = rad2deg(abs(data_pad(2,tempX2,sample_f,2))); %REMOVE PADS
Y.omega = rad2deg(abs(data_pad(2,tempY2,sample_f,2)));
Z.omega = rad2deg(abs(data_pad(2,tempZ2,sample_f,2)));

%Convert Angles to Degrees
X.theta = rad2deg(X.theta);
Y.theta = rad2deg(Y.theta);
Z.theta = rad2deg(Z.theta);

%-----
%Plot ANGLE and ANGULAR VELOCITY *****
if plotData > 0
    motion_plot(X,Y,Z,filename,1,'');
end
```

```

function data_padded = data_pad(funcnt,indata,pad_size,datatype)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014

%FUNCTION: Adds or removes padding points (reflection) from the start or
%end of a data set.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if funcnt == 1
    %ADDS "pad_size" NUMBER OF POINTS AT THE BEGINNING AND END OF THE TRIAL BY REFLECTION ✓
    data_padded = zeros(size(indata,1)+2*pad_size,size(indata,2));

    for i = pad_size:-1:1
        data_padded(i,:) = indata(1,:)-(indata(-i+pad_size+2,:)-indata(1,:));
    end
    %Add real data to padded data matrix
    for j = (pad_size+1):(length(indata)+pad_size)
        data_padded(j,:) = indata(j-pad_size,:);
    end
    %Post-data
    for k = (length(indata)+pad_size+1):(length(indata)+(2*pad_size))
        data_padded(k,:) = indata(end,:)+(indata(end,:)-indata(-k+2*length(indata)+pad_size,:)); ✓
    end

elseif funcnt == 2
    %REMOVES THE PADDING POINTS
    if datatype == 1 %EMG
        data_padded = indata(pad_size+1:length(indata)-pad_size,:);
    elseif datatype == 2 %KINEMATICS (with differentiation)
        data_padded = indata(pad_size-1:length(indata)-pad_size,:); %accounts for loss of frames in differentiation ✓
    end
end

```

```

function motion_plot(X,Y,Z,filename,plotstage,extravar)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Plots motion values (neck angle and angular velocity), trial
%midpoint, and movement phases markers (depending on input indices).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%If plotstage is 2, extravar is "trialmid" (the trial midpoint)
%If plotstage is 3, extravar is "static" (movement phases markers)

%ISOLATE TRIAL TYPE CHARACTERISTICS FROM FILENAME *****
dashloc = strfind(filename, '-');
posture = filename(dashloc(2)+1:end-4);

if plotstage == 1
    %PLOT ANGLE AND ANGULAR VELOCITY VALUES %%%%%%%%%
    index = 1:1000;
    %    index = index/50; %to plot in seconds
    figSize = get(0, 'Screensize');
    figure('Position',[figSize(3)*0.25 figSize(4)*0.10 figSize(3)*0.5 figSize(4)-
figSize(4)*0.20], 'Name', filename);
    hold on;

    %PLOT ANGLE DATA
    subplot(2,1,1)
    %    plot(index, (X.theta), 'g-', index, (Y.theta), 'b-', index, (Z.theta), 'r-', 'LineWidth',
2);
    plot(index, (X.theta), 'k:', index, (Y.theta), 'b-', index, (Z.theta), 'r--', 'LineWidth',
2);
    % use "area" to create shaded area
    xlabel('Frame Number')
    %    xlabel('Time (seconds)')
    ylabel('Angle (degrees)')
    legend('Lateral Bend', 'Axial Rotation', 'Flexion-
Extension', 'Location', 'EastOutside');
    %    line([500 500], [min(min([X.theta Y.theta Z.theta])) max(max([X.theta Y.theta Z.
theta]))])

    %PLOT ANGULAR VELOCITY DATA
    subplot(2,1,2)
    %    plot(index, X.omega, 'g-', index, Y.omega, 'b-', index, Z.omega, 'r-', 'LineWidth', 2);
    plot(index, X.omega, 'k:', index, Y.omega, 'b-', index, Z.omega, 'r--', 'LineWidth', 2);
    %    xlabel('Frame Number')
    xlabel('Time (seconds)')
    ylabel('Angular Velocity (degrees/second)')
    legend('Lateral Bend', 'Axial Rotation', 'Flexion-
Extension', 'Location', 'EastOutside');
    line([0 1000], [3 3]); %THRESHOLD CHECK LINES
    %    line([0 1000], [4 4]);
    %    line([0 1000], [5 5]);
    %    line([0 1000], [6 6]);
    %    line([0 1000], [8 8]);

```

```

% line([0 1000],[10 10]);
end

if plotstage == 2
    %PLOT PEAK ANGULAR VELOCITY LINES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    hold on;
    subplot(2,1,1)
    if strcmp(posture,'20lat') == 1
        % line([X.peak1loc X.peak1loc],[min(min([X.theta Y.theta Z.theta])) max(max(
        ([X.theta Y.theta Z.theta]))),'LineWidth',2,'LineStyle',':','Color','black')
        % line([X.peak2loc+extravar X.peak2loc+extravar],[min(min([X.theta Y.theta Z.
        theta])) max(max([X.theta Y.theta Z.theta]))),'LineWidth',
        2,'LineStyle',':','Color','black')
        elseif strcmp(posture,'45rot') == 1
            line([Y.peak1loc Y.peak1loc],[min(min([X.theta Y.theta Z.theta])) max(max([X.
            theta Y.theta Z.theta]))),'LineWidth',2,'LineStyle',':','Color','black')
            line([Y.peak2loc+extravar Y.peak2loc+extravar],[min(min([X.theta Y.theta Z.
            theta])) max(max([X.theta Y.theta Z.theta]))),'LineWidth',
            2,'LineStyle',':','Color','black')
        elseif strcmp(posture,'neutral') == 1
        else
            % line([Z.peak1loc Z.peak1loc],[min(min([X.theta Y.theta Z.theta])) max(max(
            ([X.theta Y.theta Z.theta]))),'LineWidth',2,'LineStyle',':','Color','black')
            % line([Z.peak2loc+extravar Z.peak2loc+extravar],[min(min([X.theta Y.theta Z.
            theta])) max(max([X.theta Y.theta Z.theta]))),'LineWidth',
            2,'LineStyle',':','Color','black')

        end
        subplot(2,1,2)
        if strcmp(posture,'20lat') == 1
            % line([X.peak1loc X.peak1loc],[0 max([X.peak1,X.peak2])],'LineWidth',
            2,'LineStyle',':','Color','black')
            % line([X.peak2loc+extravar X.peak2loc+extravar],[0 max([X.peak1,X.
            peak2])],'LineWidth',2,'LineStyle',':','Color','black')
            elseif strcmp(posture,'45rot') == 1
                line([Y.peak1loc Y.peak1loc],[0 max([Y.peak1,Y.peak2])],'LineWidth',
                2,'LineStyle',':','Color','black')
                line([Y.peak2loc+extravar Y.peak2loc+extravar],[0 max([Y.peak1,Y.
                peak2])],'LineWidth',2,'LineStyle',':','Color','black')
            elseif strcmp(posture,'neutral') == 1
            else
                % line([Z.peak1loc Z.peak1loc],[0 max([Z.peak1,Z.peak2])],'LineWidth',
                2,'LineStyle',':','Color','black')
                % line([Z.peak2loc+extravar Z.peak2loc+extravar],[0 max([Z.peak1,Z.
                peak2])],'LineWidth',2,'LineStyle',':','Color','black')
            end
        end

    if plotstage == 3
        %PLOT MOVEMENT PHASE ENDPOINT LINES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
        hold on;
        subplot(2,1,1)
        % extravar.A = extravar.A/50;
        % extravar.B = extravar.B/50;
        % extravar.C = extravar.C/50;

```

```

% extravar.D = extravar.D/50;
    hold on;
    minval = min(min([X.theta Y.theta Z.theta]));
    maxval = max(max([X.theta Y.theta Z.theta]));
    if extravar.A ~= extravar.B
%         rectangle('Position',[extravar.A,minval,extravar.B-extravar.A,maxval-
minval],'FaceColor',[1 1 .8])
    end
    if extravar.C ~= extravar.D
%         rectangle('Position',[extravar.C,minval,extravar.D-extravar.C,maxval-
minval],'FaceColor',[1 1 .8])
    end
%     area([0 0],[0 0]); %requires to use "alpha" command (doesn't work with
rectangle)
%     alpha(0.5)
%     line([extravar.A extravar.A],[minval maxval],'LineWidth',
2,'LineStyle','-','Color','black')
%     line([extravar.B extravar.B],[minval maxval],'LineWidth',
2,'LineStyle','-','Color','black')
%     line([extravar.C extravar.C],[minval maxval],'LineWidth',
2,'LineStyle','-','Color','black')
%     line([extravar.D extravar.D],[minval maxval],'LineWidth',
2,'LineStyle','-','Color','black')
    subplot(2,1,2)
    if strcmp(posture,'20lat') == 1
%         rectangle('Position',[extravar.A,0,extravar.B-extravar.A,max([X.peak1,X.
peak2])],'FaceColor',[1 1 .8])
        if extravar.C ~= extravar.D
%             rectangle('Position',[extravar.C,0,extravar.D-extravar.C,max([X.peak1,X.
peak2])],'FaceColor',[1 1 .8])
        end
%             line([extravar.A extravar.A],[0 max([X.peak1,X.peak2])],'LineWidth',
2,'LineStyle','-','Color','black')
%             line([extravar.B extravar.B],[0 max([X.peak1,X.peak2])],'LineWidth',
2,'LineStyle','-','Color','black')
%             line([extravar.C extravar.C],[0 max([X.peak1,X.peak2])],'LineWidth',
2,'LineStyle','-','Color','black')
%             line([extravar.D extravar.D],[0 max([X.peak1,X.peak2])],'LineWidth',
2,'LineStyle','-','Color','black')
        elseif strcmp(posture,'45rot') == 1
            rectangle('Position',[extravar.A,0,extravar.B-extravar.A,max([Y.peak1,Y.
peak2])],'FaceColor',[1 1 .8])
            if extravar.C ~= extravar.D
                rectangle('Position',[extravar.C,0,extravar.D-extravar.C,max([Y.peak1,Y.
peak2])],'FaceColor',[1 1 .8])
            end
%                 line([extravar.A extravar.A],[0 max([Y.peak1,Y.peak2])],'LineWidth',
2,'LineStyle','-','Color','black')
%                 line([extravar.B extravar.B],[0 max([Y.peak1,Y.peak2])],'LineWidth',
2,'LineStyle','-','Color','black')
%                 line([extravar.C extravar.C],[0 max([Y.peak1,Y.peak2])],'LineWidth',
2,'LineStyle','-','Color','black')
%                 line([extravar.D extravar.D],[0 max([Y.peak1,Y.peak2])],'LineWidth',
2,'LineStyle','-','Color','black')
            elseif strcmp(posture,'neutral') == 1

```



```
else
%       rectangle('Position',[extravar.A,0,extravar.B-extravar.A,max([Z.peak1,Z.
peak2])), 'FaceColor',[1 1 .8])
    if extravar.C ~= extravar.D
%       rectangle('Position',[extravar.C,0,extravar.D-extravar.C,max([Z.peak1,Z.
peak2])), 'FaceColor',[1 1 .8])
        end
%       line([extravar.A extravar.A],[0 max([Z.peak1,Z.peak2])], 'LineWidth',
2, 'LineStyle', '-', 'Color', 'black')
%       line([extravar.B extravar.B],[0 max([Z.peak1,Z.peak2])], 'LineWidth',
2, 'LineStyle', '-', 'Color', 'black')
%       line([extravar.C extravar.C],[0 max([Z.peak1,Z.peak2])], 'LineWidth',
2, 'LineStyle', '-', 'Color', 'black')
%       line([extravar.D extravar.D],[0 max([Z.peak1,Z.peak2])], 'LineWidth',
2, 'LineStyle', '-', 'Color', 'black')
    end
end
```

```

function phaseloc = motion_phases(X,Y,Z,phase_corrections,posture,filename,plotData,M)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Reads in trial midpoint and angular velocity adjustment values
%via "phase_corrections". Calculates the peak and peak location for XYZ
%angular velocity in both the first and second half of the trial, and
%identifies the "static hold" phase of the trial. The angular velocity
%threshold defines the static phase in the middle of the trial. When
%the angular velocity drops below the threshold, the static phase begins.
%When the angular velocity rises above the threshold again, the static
%phase ends.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Default Trial Midpoint
m_default = length(X.omega)/2;
%Default Angular Velocity Threshold
v_default = 3; %(degrees/second)

%CORRECT FOR TRIAL MIDPOINT ERRORS *****
%Midpoint must be between the peak angular velocities at initiation and
%return phases of the movement trial.
if phase_corrections(1) == 0 %NORMAL CASE
    trialmid = m_default;
elseif phase_corrections(1) == 1001 %NO RETURN PHASE IN TRIAL
    trialmid = length(X.omega);
    static.C = 1000;
    static.D = 1000;
elseif phase_corrections(1) == 1000 %INCOMPLETE RETURN PHASE IN TRIAL
    trialmid = m_default;
    static.D = 1000;
else %USER-SPECIFIED MID-TRIAL POINT
    trialmid = phase_corrections(1);
end

%CORRECT FOR OMEGA THRESHOLD ERRORS *****
if phase_corrections(2) == 0 %NORMAL CASE (2 DEG/SEC)
    v_threshold = v_default;
else %USER-SPECIFIED OMEGA THRESHOLD
    v_threshold = phase_corrections(2);
end

%FIND PEAK VELOCITY LOCATIONS IN FIRST AND SECOND HALF OF TRIAL *****
[X.peak1,X.peak1loc] = max(X.omega(1:trialmid));
[Y.peak1,Y.peak1loc] = max(Y.omega(1:trialmid));
[Z.peak1,Z.peak1loc] = max(Z.omega(1:trialmid));
[X.peak2,X.peak2loc] = max(X.omega(trialmid:end));
[Y.peak2,Y.peak2loc] = max(Y.omega(trialmid:end));
[Z.peak2,Z.peak2loc] = max(Z.omega(trialmid:end));

%Special case for one file that was messed up
if strcmp(M,'UW010') == 1
    if strcmp(filename,'122-hOnly-45rot30ext.CSV') == 1

```

```

[Z.peak1,Z.peak1loc] = max(Z.omega(50:trialmid));
Z.peak1loc = Z.peak1loc + 50;
end
end

%Error correction if NO RETURN PHASE
if trialmid > 999
    X.peak2loc = 1; Y.peak2loc = 1; Z.peak2loc = 1;
    trialmid = 999;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% FIND PHASE CHANGE LOCATIONS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

switch posture
    %LATERAL BEND CASE *****
    case {4} %20lat
        for h = 1:X.peak1loc %FIND INITIATION OF MOVEMENT
            if X.omega(h) > v_threshold %When signal rises above threshold
                static.A = h;
                break
            end
        end
        for i = X.peak1loc:trialmid %FIND START OF STATIC PHASE
            if X.omega(i) < v_threshold %When signal drops below threshold
                static.B = i;
                break
            end
        end
        for j = trialmid+1:(trialmid+X.peak2loc) %FIND END OF STATIC PHASE
            if X.omega(j) > v_threshold %When signal rises above threshold
                static.C = j;
                break
            end
        end
        for k = (trialmid+X.peak2loc):length(X.omega) %FIND END OF MOVEMENT
            if X.omega(k) < v_threshold %When signal rises above threshold
                static.D = k;
                break
            end
        end
    %ROTATION CASE *****
    case {5} %45rot
        for h = 1:Y.peak1loc %FIND INITIATION OF MOVEMENT
            if Y.omega(h) > v_threshold %When signal drops below threshold
                static.A = h;
                break
            end
        end
        for i = Y.peak1loc:trialmid %FIND START OF STATIC PHASE
            if Y.omega(i) < v_threshold %When signal drops below threshold
                static.B = i;
                break
            end
        end
    end
end

```

```

for j = trialmid+1:(trialmid+Y.peak2loc) %FIND END OF STATIC PHASE
    if Y.omega(j) > v_threshold %When signal rises above threshold
        static.C = j;
        break
    end
end
for k = (trialmid+Y.peak2loc):length(Y.omega) %FIND END OF MOVEMENT
    if Y.omega(k) < v_threshold %When signal rises above threshold
        static.D = k;
        break
    end
end
%FLEXION-EXTENSION CASE *****
case {2,3} %45flex,30ext
    for h = 1:Z.peak1loc %FIND INITIATION OF MOVEMENT
        if Z.omega(h) > v_threshold %When signal drops below threshold
            static.A = h;
            break
        end
    end
    for i = Z.peak1loc:trialmid %FIND START OF STATIC PHASE
        if Z.omega(i) < v_threshold %When signal drops below threshold
            static.B = i;
            break
        end
    end
    for j = trialmid+1:(trialmid+Z.peak2loc) %FIND END OF STATIC PHASE
        if Z.omega(j) > v_threshold %When signal rises above threshold
            static.C = j;
            break
        end
    end
    for k = (trialmid+Z.peak2loc):length(Z.omega) %FIND END OF MOVEMENT
        if Z.omega(k) < v_threshold %When signal rises above threshold
            static.D = k;
            break
        end
    end
case {6,7} %45rot45flex,45rot30ext
    for h = 1:Y.peak1loc %FIND INITIATION OF MOVEMENT
        if Y.omega(h) > v_threshold %When signal drops below threshold
            static.A = h;
            break
        end
    end
    for i = Z.peak1loc:trialmid %FIND START OF STATIC PHASE
        if Z.omega(i) < v_threshold %When signal drops below threshold
            static.B = i;
            break
        end
    end
    for j = trialmid+1:(trialmid+Z.peak2loc) %FIND END OF STATIC PHASE
        if Z.omega(j) > v_threshold %When signal rises above threshold
            static.C = j;
            break

```

```

        end
    end
    for k = (trialmid+Y.peak2loc):length(Y.omega) %FIND END OF MOVEMENT
        if Y.omega(k) < v_threshold %When signal rises above threshold
            static.D = k;
            break
        end
    end
    %NEUTRAL CASE (NO PHASES) *****
    case{1} %neutral
        static.A = 1;
        static.B = 1;
        static.C = 1000;
        static.D = 1000;
    end

    phaseloc = [static.A static.B static.C static.D];

    %-----
    %Plot MOVEMENT PHASES *****
    if plotData > 0
        motion_plot(X,Y,Z,filename,2,trialmid);
    end
    if plotData > 1
        motion_plot(X,Y,Z,filename,3,static);
    end
end

```

```

function [angle_peak,angle_rom] = motion_summarize(X,Y,Z,phaseloc,posture)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION:
%Reads in angle data and movement phase endpoints, and summarizes data into
%multiple output matrices (peak angle and ROM).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%All outputs in the form [Phase1: XYZ, Phase2: XYZ, Phase3: XYZ, Phase4: XYZ, Phase5:
XYZ]

if posture == 1 %neutral case

    %1)Peak Neck Angle
    angle_peak = [0 0 0 0 0 0 ...
        max(X.theta(phaseloc(2):phaseloc(3),:)) max(Y.theta(phaseloc(2):phaseloc
(3),:)) max(Z.theta(phaseloc(2):phaseloc(3),:)) ...
        0 0 0 0 0 0];
    %2)Neck Angle Range of Motion
    angle_rom = abs(angle_peak- ...
        [0 0 0 0 0 0 ...
        min(X.theta(phaseloc(2):phaseloc(3),:)) min(Y.theta(phaseloc(2):phaseloc
(3),:)) min(Z.theta(phaseloc(2):phaseloc(3),:)) ...
        0 0 0 0 0 0]);

elseif posture == 2 %45flex

    %1)Peak Neck Angle
    angle_peak = [max(X.theta(1:phaseloc(1),:)) max(Y.theta(1:phaseloc(1),:)) max(Z.
theta(1:phaseloc(1),:)) ...
        max(X.theta(phaseloc(1):phaseloc(2),:)) max(Y.theta(phaseloc(1):phaseloc
(2),:)) max(Z.theta(phaseloc(1):phaseloc(2),:)) ...
        max(X.theta(phaseloc(2):phaseloc(3),:)) max(Y.theta(phaseloc(2):phaseloc
(3),:)) max(Z.theta(phaseloc(2):phaseloc(3),:)) ...
        max(X.theta(phaseloc(3):phaseloc(4),:)) max(Y.theta(phaseloc(3):phaseloc
(4),:)) max(Z.theta(phaseloc(3):phaseloc(4),:)) ...
        max(X.theta(phaseloc(4):end,:)) max(Y.theta(phaseloc(4):end,:)) max(Z.theta
(phaseloc(4):end,:))];
    %2)Neck Angle Range of Motion
    angle_rom = abs(angle_peak- ...
        [min(X.theta(1:phaseloc(1),:)) min(Y.theta(1:phaseloc(1),:)) min(Z.theta(1:
phaseloc(1),:)) ...
        min(X.theta(phaseloc(1):phaseloc(2),:)) min(Y.theta(phaseloc(1):phaseloc
(2),:)) min(Z.theta(phaseloc(1):phaseloc(2),:)) ...
        min(X.theta(phaseloc(2):phaseloc(3),:)) min(Y.theta(phaseloc(2):phaseloc
(3),:)) min(Z.theta(phaseloc(2):phaseloc(3),:)) ...
        min(X.theta(phaseloc(3):phaseloc(4),:)) min(Y.theta(phaseloc(3):phaseloc
(4),:)) min(Z.theta(phaseloc(3):phaseloc(4),:)) ...
        min(X.theta(phaseloc(4):end,:)) min(Y.theta(phaseloc(4):end,:)) min(Z.theta
(phaseloc(4):end,:))]);

elseif posture == 3 %30ext

```

```

%1)Peak Neck Angle
angle_peak = [max(X.theta(1:phaseloc(1),:)) max(Y.theta(1:phaseloc(1),:)) min(Z.✓
theta(1:phaseloc(1),:)) ...
max(X.theta(phaseloc(1):phaseloc(2),:)) max(Y.theta(phaseloc(1):phaseloc✓
(2),:)) min(Z.theta(phaseloc(1):phaseloc(2),:)) ...
max(X.theta(phaseloc(2):phaseloc(3),:)) max(Y.theta(phaseloc(2):phaseloc✓
(3),:)) min(Z.theta(phaseloc(2):phaseloc(3),:)) ...
max(X.theta(phaseloc(3):phaseloc(4),:)) max(Y.theta(phaseloc(3):phaseloc✓
(4),:)) min(Z.theta(phaseloc(3):phaseloc(4),:)) ...
max(X.theta(phaseloc(4):end,:)) max(Y.theta(phaseloc(4):end,:)) min(Z.theta✓
(phaseloc(4):end,:))];
%2)Neck Angle Range of Motion
angle_rom = abs(angle_peak- ...
[min(X.theta(1:phaseloc(1),:)) min(Y.theta(1:phaseloc(1),:)) max(Z.theta(1:✓
phaseloc(1),:)) ...
min(X.theta(phaseloc(1):phaseloc(2),:)) min(Y.theta(phaseloc(1):phaseloc✓
(2),:)) max(Z.theta(phaseloc(1):phaseloc(2),:)) ...
min(X.theta(phaseloc(2):phaseloc(3),:)) min(Y.theta(phaseloc(2):phaseloc✓
(3),:)) max(Z.theta(phaseloc(2):phaseloc(3),:)) ...
min(X.theta(phaseloc(3):phaseloc(4),:)) min(Y.theta(phaseloc(3):phaseloc✓
(4),:)) max(Z.theta(phaseloc(3):phaseloc(4),:)) ...
min(X.theta(phaseloc(4):end,:)) min(Y.theta(phaseloc(4):end,:)) max(Z.theta✓
(phaseloc(4):end,:))]);

elseif posture == 4 %20lat

%1)Peak Neck Angle
angle_peak = [max(X.theta(1:phaseloc(1),:)) max(Y.theta(1:phaseloc(1),:)) max(Z.✓
theta(1:phaseloc(1),:)) ...
max(X.theta(phaseloc(1):phaseloc(2),:)) max(Y.theta(phaseloc(1):phaseloc✓
(2),:)) max(Z.theta(phaseloc(1):phaseloc(2),:)) ...
max(X.theta(phaseloc(2):phaseloc(3),:)) max(Y.theta(phaseloc(2):phaseloc✓
(3),:)) max(Z.theta(phaseloc(2):phaseloc(3),:)) ...
max(X.theta(phaseloc(3):phaseloc(4),:)) max(Y.theta(phaseloc(3):phaseloc✓
(4),:)) max(Z.theta(phaseloc(3):phaseloc(4),:)) ...
max(X.theta(phaseloc(4):end,:)) max(Y.theta(phaseloc(4):end,:)) max(Z.theta✓
(phaseloc(4):end,:))];

%2)Neck Angle Range of Motion
angle_rom = abs(angle_peak- ...
[min(X.theta(1:phaseloc(1),:)) min(Y.theta(1:phaseloc(1),:)) min(Z.theta(1:✓
phaseloc(1),:)) ...
min(X.theta(phaseloc(1):phaseloc(2),:)) min(Y.theta(phaseloc(1):phaseloc✓
(2),:)) min(Z.theta(phaseloc(1):phaseloc(2),:)) ...
min(X.theta(phaseloc(2):phaseloc(3),:)) min(Y.theta(phaseloc(2):phaseloc✓
(3),:)) min(Z.theta(phaseloc(2):phaseloc(3),:)) ...
min(X.theta(phaseloc(3):phaseloc(4),:)) min(Y.theta(phaseloc(3):phaseloc✓
(4),:)) min(Z.theta(phaseloc(3):phaseloc(4),:)) ...
min(X.theta(phaseloc(4):end,:)) min(Y.theta(phaseloc(4):end,:)) min(Z.theta✓
(phaseloc(4):end,:))]);

elseif posture == 5 %45rot

%1)Peak Neck Angle

```

```

    angle_peak = [min(X.theta(1:phaseloc(1),:)) max(Y.theta(1:phaseloc(1),:)) max(Z.✓
theta(1:phaseloc(1),:)) ...
        min(X.theta(phaseloc(1):phaseloc(2),:)) max(Y.theta(phaseloc(1):phaseloc✓
(2),:)) max(Z.theta(phaseloc(1):phaseloc(2),:)) ...
        min(X.theta(phaseloc(2):phaseloc(3),:)) max(Y.theta(phaseloc(2):phaseloc✓
(3),:)) max(Z.theta(phaseloc(2):phaseloc(3),:)) ...
        min(X.theta(phaseloc(3):phaseloc(4),:)) max(Y.theta(phaseloc(3):phaseloc✓
(4),:)) max(Z.theta(phaseloc(3):phaseloc(4),:)) ...
        min(X.theta(phaseloc(4):end,:)) max(Y.theta(phaseloc(4):end,:)) max(Z.theta✓
(phaseloc(4):end,:))];
    %2)Neck Angle Range of Motion
    angle_rom = abs(angle_peak- ...
        [max(X.theta(1:phaseloc(1),:)) min(Y.theta(1:phaseloc(1),:)) min(Z.theta(1:✓
phaseloc(1),:)) ...
        max(X.theta(phaseloc(1):phaseloc(2),:)) min(Y.theta(phaseloc(1):phaseloc✓
(2),:)) min(Z.theta(phaseloc(1):phaseloc(2),:)) ...
        max(X.theta(phaseloc(2):phaseloc(3),:)) min(Y.theta(phaseloc(2):phaseloc✓
(3),:)) min(Z.theta(phaseloc(2):phaseloc(3),:)) ...
        max(X.theta(phaseloc(3):phaseloc(4),:)) min(Y.theta(phaseloc(3):phaseloc✓
(4),:)) min(Z.theta(phaseloc(3):phaseloc(4),:)) ...
        max(X.theta(phaseloc(4):end,:)) min(Y.theta(phaseloc(4):end,:)) min(Z.theta✓
(phaseloc(4):end,:))]);

elseif posture == 7 %45rot45ext

    %1)Peak Neck Angle
    angle_peak = [min(X.theta(1:phaseloc(1),:)) max(Y.theta(1:phaseloc(1),:)) min(Z.✓
theta(1:phaseloc(1),:)) ...
        min(X.theta(phaseloc(1):phaseloc(2),:)) max(Y.theta(phaseloc(1):phaseloc✓
(2),:)) min(Z.theta(phaseloc(1):phaseloc(2),:)) ...
        min(X.theta(phaseloc(2):phaseloc(3),:)) max(Y.theta(phaseloc(2):phaseloc✓
(3),:)) min(Z.theta(phaseloc(2):phaseloc(3),:)) ...
        min(X.theta(phaseloc(3):phaseloc(4),:)) max(Y.theta(phaseloc(3):phaseloc✓
(4),:)) min(Z.theta(phaseloc(3):phaseloc(4),:)) ...
        min(X.theta(phaseloc(4):end,:)) max(Y.theta(phaseloc(4):end,:)) min(Z.theta✓
(phaseloc(4):end,:))];
    %2)Neck Angle Range of Motion
    angle_rom = abs(angle_peak- ...
        [max(X.theta(1:phaseloc(1),:)) min(Y.theta(1:phaseloc(1),:)) max(Z.theta(1:✓
phaseloc(1),:)) ...
        max(X.theta(phaseloc(1):phaseloc(2),:)) min(Y.theta(phaseloc(1):phaseloc✓
(2),:)) max(Z.theta(phaseloc(1):phaseloc(2),:)) ...
        max(X.theta(phaseloc(2):phaseloc(3),:)) min(Y.theta(phaseloc(2):phaseloc✓
(3),:)) max(Z.theta(phaseloc(2):phaseloc(3),:)) ...
        max(X.theta(phaseloc(3):phaseloc(4),:)) min(Y.theta(phaseloc(3):phaseloc✓
(4),:)) max(Z.theta(phaseloc(3):phaseloc(4),:)) ...
        max(X.theta(phaseloc(4):end,:)) min(Y.theta(phaseloc(4):end,:)) max(Z.theta✓
(phaseloc(4):end,:))]);

elseif posture == 6 %45rot30flex

    %1)Peak Neck Angle
    angle_peak = [min(X.theta(1:phaseloc(1),:)) max(Y.theta(1:phaseloc(1),:)) max(Z.✓
theta(1:phaseloc(1),:)) ...
        min(X.theta(phaseloc(1):phaseloc(2),:)) max(Y.theta(phaseloc(1):phaseloc✓

```



```

(2,:) max(Z.theta(phaseloc(1):phaseloc(2),:)) ...
      min(X.theta(phaseloc(2):phaseloc(3),:)) max(Y.theta(phaseloc(2):phaseloc(3),:)) ✓
(3,:) max(Z.theta(phaseloc(2):phaseloc(3),:)) ...
      min(X.theta(phaseloc(3):phaseloc(4),:)) max(Y.theta(phaseloc(3):phaseloc(4),:)) ✓
(4,:) max(Z.theta(phaseloc(3):phaseloc(4),:)) ...
      min(X.theta(phaseloc(4):end,:)) max(Y.theta(phaseloc(4):end,:)) max(Z.theta(phaseloc(4):end,:)) ✓
(phaseloc(4):end,:)]];

%2) Neck Angle Range of Motion
angle_rom = abs(angle_peak - ...
    [max(X.theta(1:phaseloc(1),:)) min(Y.theta(1:phaseloc(1),:)) min(Z.theta(1:phaseloc(1),:)) ...
    max(X.theta(phaseloc(1):phaseloc(2),:)) min(Y.theta(phaseloc(1):phaseloc(2),:)) min(Z.theta(phaseloc(1):phaseloc(2),:)) ...
    max(X.theta(phaseloc(2):phaseloc(3),:)) min(Y.theta(phaseloc(2):phaseloc(3),:)) min(Z.theta(phaseloc(2):phaseloc(3),:)) ...
    max(X.theta(phaseloc(3):phaseloc(4),:)) min(Y.theta(phaseloc(3):phaseloc(4),:)) min(Z.theta(phaseloc(3):phaseloc(4),:)) ...
    max(X.theta(phaseloc(4):end,:)) min(Y.theta(phaseloc(4):end,:)) min(Z.theta(phaseloc(4):end,:))]);

end

%CORRECT FOR MOTION PHASE ERRORS
if phaseloc(1) == 1 %NO PHASE 1
    angle_peak(1:3) = 0;
    angle_rom(1:3) = 0;
end
if phaseloc(2) == phaseloc(1) %NO PHASE 2
    angle_peak(4:6) = 0;
    angle_rom(4:6) = 0;
end
if phaseloc(3) == 1000 %NO RETURN PHASE
    angle_peak(10:12) = 0;
    angle_rom(10:12) = 0;
end
if phaseloc(4) == 1000 %INCOMPLETE RETURN PHASE
    angle_peak(13:15) = 0;
    angle_rom(13:15) = 0;
end
end

```



```
function track_progress(N)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Provides output to track process of data processing.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Track Processing Progress *****
if N == 10 || N == 20 || N == 30 || N == 40 || N == 50 || N == 60 || N == 70 || N == 80 || N == 90 || N == 100 || N == 110 || N == 120
    fprintf([num2str(N) '\n'])
else
    fprintf('.')
end
```

Appendix A4: MATLAB code for processing EMG data.

```

function EMG_PROCESS(filetype,startfile,plotData,outData,plotOnset)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION:
%Reads in electromyography (EMG) data files. For MVE trials, data is linear
%enveloped, and maximum amplitudes for each muscle across all trials are
%extracted. For EXP trials, data is linear enveloped, normalized to MVE,
%and several summary values are extracted from each trial
%(see emg_trial_summarize.m). All summary values are exported to an Excel
%spreadsheet (.xlsx).

%INPUTS:
%filetype          String indicator of file type (EXP or MVE)
%startfile         Integer which indicates the trial number to start with.
%plotData          Integer which indicates plot characteristics for each
%                  trial.
%outData           Integer which indicates whether or not to output data from
%                  processed trials. 0 = no output; 1 = output.
%plotOnset         Integer which indicates whether or not to apply onset of
%                  activity points to output plot.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc;
tstart = tic;

%Confirm Correct "filetype" Input *****
if strcmp(filetype,'MVE') == 0 && strcmp(filetype,'EXP') == 0
    fprintf('INCORRECT INPUT PARAMETER: Input must be "MVE" or "EXP".\n')
    return;
else
    info.filetype = filetype;
end

%-----
%FUNCTION PARAMETERS AND SETTINGS %%%%%%%%%%%%%%%
%Data Structure and Input Parameters *****
cd ..
cd ..
info.root = [cd '\Data\']; %Root Data Folder
cd('Analysis\EMG')
info.subjects = [{'UW002'},{'UW003'},{'UW005'},{'UW006'},{'UW007'},{'UW008'},↵
{'UW009'},{'UW010'}]; %Participant Identification Codes
% info.subjects = {'UW007'};
info.downsamp = 30; %Downsampling factor for data (i.e. 30 = grab every 30th frame, or↵
1500Hz -> 50Hz)
info.sample_f = 1500; %Data Collection Sample Frequency (Hz)
info.sample_d = info.sample_f/info.downsamp; %Down-Sampled Sample Frequency (Hz)

%EMG Filter Characteristics *****
smooth.order = 2.0; %Filter Order
smooth.cutoff = 1; %Filter Cutoff Frequency (Hz)
smooth.filt_type = 'low'; %Filter Type: 'high' OR 'low' pass

```

[illegible]

```

    emg_plot(info.sample_d,emgdata_mve,plotData,'Time (seconds)','EMG'
Amplitude (% MVE)',[pathname((end-14):end) filename{N}]);
    %Mark Movement Phases on Plot
    if static(1) ~= static(2)
        rectangle('Position',[static(1)/info.sample_d,min(min
(emgdata_mve)),static(2)/info.sample_d-static(1)/info.sample_d,max(max(emgdata_mve))-
min(min(emgdata_mve))],'FaceColor',[1 1 .8])
    end
    if static(3) ~= static(4)
        rectangle('Position',[static(3)/info.sample_d,min(min
(emgdata_mve)),static(4)/info.sample_d-static(3)/info.sample_d,max(max(emgdata_mve))-
min(min(emgdata_mve))],'FaceColor',[1 1 .8])
    end
    area([0 0],[0 0]); %requires to use "alpha" command (doesn't work with
rectangle)
    alpha(0.5)
end

%Calculate Summary Data For the Current File *****
[maxtime{N},onsettime{N},amp_peak{N},amp_mean{N},amp_medn{N},amp_rms{N},
amp_weight{N},amp_apdf{N}] ...
    = emg_trial_summarize(emgdata_mve,static,posture{N},plotData,
plotOnset);
end

%-----
%DATA EXPORT (BY TRIAL) >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
%
    if outData == 1
        if strcmp filetype,'EXP') == 1
            dlmwrite([info.root info.subjects{M} '\Processed EMG\permve_' filename
{N}(1:end-4) '.csv'],emgdata_mve); %EXPORT: Processed EMG signals (%MVE)
        end
        dlmwrite([info.root info.subjects{M} '\Processed EMG\volts_' filename{N}
(1:end-4) '.csv'],emgdata{N}); %EXPORT: Processed EMG signals (Volts)
    end
    track_progress(N)
end %FILE

%-----
%DATA EXPORT (BY PARTICIPANT) >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
if outData == 1
    if strcmp filetype,'MVE') == 1
        emg_mve_summarize(numfiles,emgdata,filename,info.subjects{M},info); %Write
MVE Data Summary File
    else
        %Write EMG SUMMARY DATA to Files
        xlswrite([info.root 'Output\summary_emg_timetomax.xlsx'],[cell2mat(subject
(startfile:numfiles)) cell2mat(trialnum(startfile:numfiles)) cell2mat(helmet
(startfile:numfiles)) cell2mat(posture(startfile:numfiles)) cell2mat(maxtime
(startfile:numfiles))],info.subjects{M},'A3');
        xlswrite([info.root 'Output\summary_emg_onset.xlsx'],[cell2mat(subject
(startfile:numfiles)) cell2mat(trialnum(startfile:numfiles)) cell2mat(helmet
(startfile:numfiles)) cell2mat(posture(startfile:numfiles)) cell2mat(onsettime
(startfile:numfiles))],info.subjects{M},'A3');
        xlswrite([info.root 'Output\summary_emg_peak.xlsx'],[cell2mat(subject

```

```

(startfile:numfiles)') cell2mat(trialnum(startfile:numfiles)') cell2mat(helmet ✓
(startfile:numfiles)') cell2mat(posture(startfile:numfiles)') cell2mat(amp_peak ✓
(startfile:numfiles)')],info.subjects{M}, 'A3');
        xlswrite([info.root 'Output\summary_emg_mean.xlsx'],[cell2mat(subject ✓
(startfile:numfiles)') cell2mat(trialnum(startfile:numfiles)') cell2mat(helmet ✓
(startfile:numfiles)') cell2mat(posture(startfile:numfiles)') cell2mat(amp_mean ✓
(startfile:numfiles)')],info.subjects{M}, 'A3');
        xlswrite([info.root 'Output\summary_emg_median.xlsx'],[cell2mat(subject ✓
(startfile:numfiles)') cell2mat(trialnum(startfile:numfiles)') cell2mat(helmet ✓
(startfile:numfiles)') cell2mat(posture(startfile:numfiles)') cell2mat(amp_medn ✓
(startfile:numfiles)')],info.subjects{M}, 'A3');
        xlswrite([info.root 'Output\summary_emg_rms.xlsx'],[cell2mat(subject ✓
(startfile:numfiles)') cell2mat(trialnum(startfile:numfiles)') cell2mat(helmet ✓
(startfile:numfiles)') cell2mat(posture(startfile:numfiles)') cell2mat(amp_rms ✓
(startfile:numfiles)')],info.subjects{M}, 'A3');
        xlswrite([info.root 'Output\summary_emg_weight.xlsx'],[cell2mat(subject ✓
(startfile:numfiles)') cell2mat(trialnum(startfile:numfiles)') cell2mat(helmet ✓
(startfile:numfiles)') cell2mat(posture(startfile:numfiles)') cell2mat(amp_weight ✓
(startfile:numfiles)')],info.subjects{M}, 'A3');
        xlswrite([info.root 'Output\summary_emg_apdf.xlsx'],[cell2mat(subject ✓
(startfile:numfiles)') cell2mat(trialnum(startfile:numfiles)') cell2mat(helmet ✓
(startfile:numfiles)') cell2mat(posture(startfile:numfiles)') cell2mat(amp_apdf ✓
(startfile:numfiles)')],info.subjects{M}, 'A3');
    end
end

disp([info.subjects{M} ' Complete.'])
toc(tstart)

end %SUBJECT

```



```
function [pathname,filename,numfiles] = file_loader(info,index)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Finds appropriate file path and filenames based on specified
%root folder and trial type variables. Returns "pathname", "filename",
%and the number of files contained in "filename".
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%SELECT FILES TO PROCESS
%Extract list of filenames for subject MVE folder
pathname = [info.root info.subjects{index} '\Raw\' info.filetype '\'];
filenames = dir(pathname(1:end-1));
filename = {filenames.name};
filename(1:2) = [];

%Determine number of active files (# of files pulled from MVE directory)
numfiles = size(filename,2);
```

```

function outdata = emg_linear_envelope(fileName,headerRow,headerCol,numCol,info, ✓
smooth,heart)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014

%FUNCTION: Runs a linear envelope process on incoming data. Process imports
%data according to filename, removes DC offset, pads data, applies a heart
%rate removal filter (optional), full wave rectifies data, and then
%smoothes data with a Butterworth digital filter.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%IMPORT FILE DATA *****
imdata = importdata(fileName, ',', headerRow); %import
emgdata = imdata.data(:,headerCol+1:end); %removes 2 header columns
%data.temp1 = emgdata;

%Remove linear DC offset (trend) from signal
emgdata = detrend(emgdata);
%data.temp2 = emgdata;

%MODIFY DATA *****
%Add 1-second of padding points to data (Howarth & Callaghan, 2009)
emgdata = data_pad(1,emgdata,info.sample_f,1);

%Construct and apply heart rate filter
if heart.status == 1
    heart.cutoff/(info.sample_f/2);
    [b,a] = butter(heart.order,heart.cutoff/(info.sample_f/2),heart.filt_type);
    emgdata = filtfilt(b,a,emgdata);
    %data.temp3 = emgdata;
end

%LINEAR ENVELOPE DATA *****
%Full wave rectification
emgdata = abs(emgdata);
%data.temp4 = emgdata;

%Construct and apply standard BW filter
smooth.cutoff/(info.sample_f/2);
[b,a] = butter(smooth.order,smooth.cutoff/(info.sample_f/2),smooth.filt_type);
emgdata = filter(b,a,emgdata);
%data.temp5 = emgdata;

%Remove extra padding points from the data
emgdata = data_pad(2,emgdata,info.sample_f,1);
%data.temp6 = emgdata;

%Remove extra analog channels and prepare EMGDATA for output
outdata_temp = emgdata(:,1:numCol);

%Down sample EMG DATA from 1500 Hz to 50 Hz to match MOTION DATA
outdata = downsample(outdata_temp,info.downsamp);

```



```

function data_padded = data_pad(funcnt,indata,pad_size,datatype)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014

%FUNCTION: Adds or removes padding points (reflection) from the start or
%end of a data set.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if funcnt == 1
    %ADDS "pad_size" NUMBER OF POINTS AT THE BEGINNING AND END OF THE TRIAL BY REFLECTION ✓
    data_padded = zeros(size(indata,1)+2*pad_size,size(indata,2));

    for i = pad_size:-1:1
        data_padded(i,:) = indata(1,:)-(indata(-i+pad_size+2,:)-indata(1,:));
    end
    %Add real data to padded data matrix
    for j = (pad_size+1):(length(indata)+pad_size)
        data_padded(j,:) = indata(j-pad_size,:);
    end
    %Post-data
    for k = (length(indata)+pad_size+1):(length(indata)+(2*pad_size))
        data_padded(k,:) = indata(end,:)+(indata(end,:)-indata(-k+2*length(indata)+pad_size,:)); ✓
    end

elseif funcnt == 2
    %REMOVES THE PADDING POINTS
    if datatype == 1 %EMG
        data_padded = indata(pad_size+1:length(indata)-pad_size,:);
    elseif datatype == 2 %KINEMATICS (with differentiation)
        data_padded = indata(pad_size-1:length(indata)-pad_size,:); %accounts for loss of frames in differentiation ✓
    end
end

```

```

function [t_num,h_num,p_num,nohelm] = file_characteristics(filename)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Break down filename into trial characteristics.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Break down filename into trial characteristics
dashloc = strfind(filename,'-');
t_str = filename(1:dashloc(1)-1); %TRIAL NUMBER ID
h_str = filename(dashloc(1)+1:dashloc(2)-1); %HELMET TYPE ID
p_str = filename(dashloc(2)+1:end-4); %POSTURE ID

%Assign numerical code to each characteristics (makes ofr easier file I/O)
%TRIAL NUMBER ID
t_num = str2num(t_str);
nohelm = 2; %designates as a helmet (not "nohelmet")

%HELMET TYPE ID
switch h_str
    case{'nohelmet'}
        h_num = 1;
        nohelm = 1;
    case{'hOnly'}
        h_num = 2;
    case{'hNVGup'}
        h_num = 3;
    case{'hNVGdown'}
        h_num = 4;
    case{'hCWup'}
        h_num = 5;
    case{'hCWdown'}
        h_num = 6;
end

%POSTURE ID
switch p_str
    case{'neutral'}
        p_num = 1;
    case{'45flex'}
        p_num = 2;
    case{'30ext'}
        p_num = 3;
    case{'20lat'}
        p_num = 4;
    case{'45rot'}
        p_num = 5;
    case{'45rot45flex'}
        p_num = 6;
    case{'45rot30ext'}
        p_num = 7;
end

```

```
function emgnorm = emg_normalize(info,M,indata)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014

%FUNCTION: Loads MVE Summary file (peak EMG amplitudes across all MVEs for
%each trial), and normalizes trials to those values.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%LOAD SESSION MVE SUMMARY VALUES
try
    mve_data = xlsread([info.root 'mve_summary.xlsx'],info.subjects{M},'A1:A10');
catch
    %Catch error if no MVE Summary file detected
    fprintf(['An MVE Summary file could not be found for ' sub_id ', and the current
file could not be normalized to MVE.\nPlease check source files or run EMG_process on
the MVE files.\n']);
    return;
end

%CONVERT TRIAL VALUES TO PERCENT OF MVE
emgnorm = bsxfun(@rdivide,indata,mve_data')*100;
```

```

function emg_plot(sample_f,data,channels,x_label,y_label,filename)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Plots channels of analog data (EMG).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if strcmp('Time (seconds)',x_label) == 1
    %Create time index (x-axis data)
    stepLength = 1/sample_f;
    index = [];
    for stepNum = 0:(size(data,1)-1)
        index = [index; stepNum*stepLength];
    end
else
    index = 1:length(data);
end

%Plot Data
style = {'k-','b-','r-','m-','g-','k-','b:','r:','m:','g:'};
% style = {'k-','b-','r-','m-','g-','k-','b-','r-','m-','g-'};
legendval = {'LS-R','LS-L','ES-R','ES-L','SC-R','SC-L','SCM-R','SCM-L','UT-R','UT-L'};
figSize = get(0,'Screensize');
figure('Position',[figSize(3)*0.13 figSize(4)*0.10 figSize(3)*0.75 figSize(4)-
figSize(4)*0.20],'Name',filename);
hold on;
title(filename,'interpreter','none')
xlabel(x_label);
ylabel(y_label);
for N = 1:length(channels)
    plot(index,data(:,channels(N)),style{channels(N)},'LineWidth',2);
    % set(gca,'XTick',1:25:1000)
end
legend(legendval(channels),'Location','EastOutside')

```

```

function [maxtime,onsettime2,amp_peak,amp_mean,amp_medn,amp_rms,amp_weight,amp_apdf] = ✓
emg_trial_summarize(emgdata,static,posture,plotData,plotOnset)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Reads in EMG data, and calculates the follwoing summary values:
%-time to maximum activity (by movement phase)
%-time to onset of activation (by movement phase)
%-peak amplitude (by movement phase)
%-mean amplitude (by movement phase)
%-median amplitude (by movement phase)
%-RMS (root mean squared) amplitude (by movement phase)
%-Weighted Euclidean Magnitude (by movement phase)
%-Amplitude probability distribution function (APDF) values (0.1, 0.5, 0.9)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%All outputs in the form [Phase1: 10 muscles, Phase2: 10 muscles, Phase3: 10 muscles, ✓
Phase4: 10 muscles, Phase5: 10 muscles]

% > TIME TO ONSET OF ACTIVATION (Prior to movement markers B, C, and D) >>>
OTL = 250; OTLf = round(OTL/1000*50); %OTL (onset time lag) = onset time constant, in ✓
ms; OTLf in frames
SAL = 250; SALf = round(SAL/1000*50); %SAL (static activity lag) = window to obtain ✓
static activity prior to onset, in ms; SALf in frames

%Caluclate Muscle Activity Onset Time
switch posture
case{1}
    onsettime2 = nan(1,30); %no phase markers, so no onset times
otherwise
    %Caluclate mean and s.d. in static activity window before each phase
    phase_mean = [mean(emgdata(static(2)-(OTLf+SALf):static(2)-OTLf,:)) mean ✓
(emgdata(static(3)-(OTLf+SALf):static(3)-OTLf,:)) mean(emgdata(static(4)-(OTLf+SALf): ✓
static(4)-OTLf,:))];
    phase_stdv = [std(emgdata(static(2)-(OTLf+SALf):static(2)-OTLf,:)) std(emgdata ✓
(static(3)-(OTLf+SALf):static(3)-OTLf,:)) std(emgdata(static(4)-(OTLf+SALf):static(4)- ✓
OTLf,:))];

    for ch = 1:10; %10 channels of EMG data
        for phase = 2:4
            if static(phase)-static(phase-1) < OTLf+SALf
                onset(phase-1) = NaN;
            else
                for x = 1:OTLf %lenth of onset time lag window
                    if emgdata(static(phase)-OTLf+x,ch) > phase_mean((phase-2) ✓
*10+ch)+2*phase_stdv((phase-2)*10+ch)
                        onset(phase-1) = (static(phase)-OTLf+x);
                        break
                    else
                        onset(phase-1) = NaN;
                    end
                end
            end
        end
    end
end

```


[illegible]

[illegible]

```
function [var1,var2,var3,var4] = fix_outputs(usize,static,var1,var2,var3,var4)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Adjust summary outputs according to common data problems.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%CORRECT FOR MOTION PHASE ERRORS *****
if static(1) == 1 %NO PHASE 1
    var1(1:usize) = NaN; %phase 1 values
end
if static(2) == static(1) %NO PHASE 2 (NEUTRAL TRIALS)
    var2(1:usize) = NaN; %phase 2 values
end
if static(3) == 1000 %NO RETURN PHASE (NO PHASE 4 OR 5)
    var3(1:usize) = NaN; %phase 4 values
end
if static(4) == 1000 %INCOMPLETE RETURN PHASE (NO PHASE 5)
    var4(1:usize) = NaN; %phase 5 values
end
```

```

function emg_plot_onset(emgdata,plotData,OTLf,SALf,static,onsettime,onsetframe)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Plots onset of activity on each channel of EMG
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

OTLp = [(static(2)-OTLf)/50 (static(3)-OTLf)/50 (static(4)-OTLf)/50];
SALp = [(static(2)-OTLf-SALf)/50 (static(3)-OTLf-SALf)/50 (static(4)-OTLf-SALf)/50];

if static(2)-static(1) > OTLf+SALf
    line([OTLp(1) OTLp(1)], [min(min(emgdata)) max(max(emgdata))], 'LineStyle',':')
    line([SALp(1) SALp(1)], [min(min(emgdata)) max(max(emgdata))], 'LineStyle',':')
end
if static(3)-static(2) > OTLf+SALf
    line([OTLp(2) OTLp(2)], [min(min(emgdata)) max(max(emgdata))], 'LineStyle',':')
    line([SALp(2) SALp(2)], [min(min(emgdata)) max(max(emgdata))], 'LineStyle',':')
end
if static(4)-static(3) > OTLf+SALf
    line([OTLp(3) OTLp(3)], [min(min(emgdata)) max(max(emgdata))], 'LineStyle',':')
    line([SALp(3) SALp(3)], [min(min(emgdata)) max(max(emgdata))], 'LineStyle',':')
end

indi = {'k+', 'b+', 'r*', 'm*', 'g*', 'kx', 'bx', 'rx', 'mx', 'gx'};
for h = 1:3
    for i = 1:length(plotData)
        if onsettime(plotData(i),h) > 0
            plot(onsetframe(plotData(i),h)/50,emgdata(onsetframe(plotData(i),h),
plotData(i)),indi{plotData(i)}, 'LineWidth',2)
        end
    end
end
end

```

```
function track_progress(N)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014
%
%FUNCTION: Provides output to track process of data processing.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Track Processing Progress *****
if N == 10 || N == 20 || N == 30 || N == 40 || N == 50 || N == 60 || N == 70 || N == 80 || N == 90 || N == 100 || N == 110 || N == 120
    fprintf([num2str(N) '\n'])
else
    fprintf('.')
end
```

```

function emg_mve_summarize(numfiles,emgdata,filename,sub_id,info)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colin D. McKinnon, UNIVERSITY OF WATERLOO
%Created in MATLAB R2014a (MathWorks, Natick, MA, USA)
%Last Modified: March 2014

%FUNCTION: Determines the maximum EMG amplitude for each muscle across all
%MVE trials, and exports those values and associated time/row indices to a
%summary file named "mve summary.xlsx".
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Calculate the peak amplitude in each column for each file
temp1 = []; temp2 = [];
for x = 1:numfiles
    [max1,max2] = max(emgdata{x}); %max1 = peak value; max2 = peak index
    temp1 = [temp1; max1]; %Create matrix of max values for each trial [trials x
muscles]
    temp2 = [temp2; max2]; %Create associated matrix of max indices in each trial
[trials x muscle]
end

%Determine the VALUE, associated ROW INDEX, and FILENAME where the max
%amplitude occurred for each of the 10 muscles
filecount = 1;
maxfilelist = [];
for h = 1:10
    [peak.value{h},temp3(h)] = max(temp1(:,h)); %Extract maximum values for each
muscle across all trials [1 x muscles]
    peak.index{h} = temp2(temp3(h),h); %Extract the row index for the trial where the
max occurred
    peak.file{h} = filename{temp3(h)}; %Extract the filename for the trial where the
max occurred
end

%Determine list of files which produced maximum values and plot channels
%with max values
for j = 1:numfiles
    temp = cellfun('isempty',strfind(peak.file,filename{j}));
    if sum(temp) < 10
        channels = find(temp == 0);
        emg_plot(info.sample_f,emgdata{j},channels,'Frame','EMG Amplitude (Volts)',
[sub_id ' \\' filename{j}]);
    end
end

%Write MVE Summary File *****
% File with maximum values for each column. Values for each file are
% represented in a row, and are paired with their file name.
columnheader = {'LS-R','LS-L','ES-R','ES-L','SC-R','SC-L','SCM-R','SCM-L','UT-R','UT-
L'};

xlswrite([info.root 'mve_summary.xlsx'], columnheader,sub_id, 'C1');
xlswrite([info.root 'mve_summary.xlsx'], peak.value,sub_id, 'A1');
xlswrite([info.root 'mve_summary.xlsx'], peak.index,sub_id, 'B1');
xlswrite([info.root 'mve_summary.xlsx'], peak.file,sub_id, 'D1');

```


Appendix A5: Full Statistical listing for EMG measures.

COLOR CODE

	Meaning
Red	Significant p-value
Yellow	Interaction
Blue	Main effect – Posture
Green	Main effect – Helmet (and could be posture too)

ABBREVIATIONS (For a full description of the measures see Page 13-14 of the report):

EMG – Electromyography

APDF – Amplitude Probability Distribution Function

RMS – Root Mean Square

MCI – Muscular Co-activation Index

NAMING CONVENTION / LEGEND

For EMG – APDF only

XX_YYY_Z

Where:

XX – is the probability from the APDF distribution (10, 50 or 90)

YYY – is the muscle:

LS = levator scapulae

ES = cervical erector spinae

SC = splenius capitis

SCM = sternocleidomastoid

UT = upper trapezius

Z – side of the body from the participant perspective (R = Right; L = Left)

For EMG – MCI only

XX

Where:

XX – trial phase (from 1 to 5 – see Page 13 for a description of the phases)

All other EMG Measures (Mean, Median, Onset, Peak, RMS, Time to Max)

XX_YYY_Z

XX – trial phase (from 1 to 5 – see Page 13 for a description of the phases)

YYY – is the muscle:

LS = levator scapulae

ES = cervical erector spinae

SC = splenius capitis

SCM = sternocleidomastoid

UT = upper trapezius

Z – side of the body from the participant perspective (R = Right; L = Left)

EMG – APDF

	Helmet	Posture	Helmet*Posture	Pairwise Comparison
10_LS_R	0.2515	0.0428	0.7638	Posture3 different from 5 (<0.05)
10_LS_L	0.3745	0.7399	0.0876	
10_ES_R	0.5435	<.0001	0.0101	Interaction
10_ES_L	0.5390	<.0001	0.0022	Interaction
10_SC_R	0.2614	0.0111	0.2850	Posture3 different from 1 (<0.05)
10_SC_L	0.0092	<.0001	0.0013	Interaction
10_SCM_R	0.0092	<.0001	0.0013	Interaction
10_SCM_L	0.3144	<.0001	0.1844	Posture7 different from all except 5 (<0.05) Posture5 different from 1 (<0.05)
10_UT_R	0.5120	0.0704	0.3566	
10_UT_L	0.4915	0.1335	0.0609	
50_LS_R	0.1737	0.0041	0.1766	Posture3 different from 1, 4, 5 (<0.05)
50_LS_L	0.2001	0.0081	0.0126	Interaction
50_ES_R	0.4559	<.0001	0.0006	Interaction
50_ES_L	0.4458	<.0001	0.0001	Interaction
50_SC_R	0.2065	0.0188	0.7829	Posture3 different from 1 and 4 (<0.05)
50_SC_L	0.0398	<.0001	0.1945	Helmet1 different from 3 (<0.05) Posture7 different from all (<0.05) Posture5 different from 1 (<0.05)
50_SCM_R	0.2599	0.0013	0.0205	Interaction
50_SCM_L	0.5903	<.0001	0.1364	Posture7 different from all (<0.05) Posture5 different from all except 3 and 6 (<0.05) Posture1 different from 3,5,7 (<0.05) Posture2,4 different from 5,7 (<0.05)
50_UT_R	0.5362	0.0648	0.4800	
50_UT_L	0.5216	0.0650	0.0868	
90_LS_R	0.1639	0.004	0.0289	Interaction
90_LS_L	0.2293	0.0108	0.0049	Interaction
90_ES_R	0.3359	<.0001	0.0014	Interaction
90_ES_L	0.4336	<.0001	0.0014	Interaction
90_SC_R	0.2293	0.0543	0.7287	
90_SC_L	0.0396	<.0001	0.1457	Helmet1 different from 3 (<0.05) Posture7 different from all (<0.01) Posture5 different from 1 (<0.05)
90_SCM_R	0.3856	0.0017	0.0143	Interaction
90_SCM_L	0.4992	<.0001	0.1718	Posture7 different from all (<0.05) Posture1 different from all except 2 and 4 Posture2 different from 5 and 7 Posture3 different from 1 and 7 Posture5 different from all except 3 and 6
90_UT_R	0.5629	0.0850	0.5535	
90_UT_L	0.6136	0.0679	0.1973	

EMG – Mean

	Helmet	Posture	Helmet*Posture	Pairwise Comparison
P1_LS_R	0.4901	0.4753	0.0115	Interaction
P1_LS_L	0.5057	0.3324	0.2456	
P1_ES_R	0.3565	0.6262	0.5692	
P1_ES_L	0.9276	0.4543	0.5989	
P1_SC_R	0.132	0.6091	0.6352	
P1_SC_L	0.9581	0.5437	0.364	
P1_SCM_R	0.6652	0.2197	0.1075	
P1_SCM_L	0.8394	0.1754	0.3639	
P1_UT_R	0.2052	0.1318	0.773	
P1_UT_L	0.9842	0.2272	0.9092	
P2_LS_R	0.2821	0.2691	0.7797	
P2_LS_L	0.1214	0.0145	<.0001	Interaction
P2_ES_R	0.5058	0.0031	0.0973	Posture6 different from 3 and 4
P2_ES_L	0.4424	0.0011	0.1701	Posture6 different from 3 and 4 Posture7 different from 3
P2_SC_R	0.3892	0.5377	0.2849	
P2_SC_L	0.0088	<.0001	0.0166	Interaction
P2_SCM_R	0.6573	0.0677	0.0419	Interaction
P2_SCM_L	0.5101	<.0001	0.4214	Posture5,6,7 different from 2,3,4
P2_UT_R	0.7378	0.5154	0.4204	
P2_UT_L	0.6906	0.0921	0.1110	
P3_LS_R	0.1423	0.0046	0.0832	
P3_LS_L	0.1969	0.0046	0.0072	Interaction
P3_ES_R	0.4315	<.0001	0.0004	Interaction
P3_ES_L	0.4197	<.0001	<.0001	Interaction
P3_SC_R	0.1790	0.0231	0.7383	
P3_SC_L	0.0716	<.0001	0.3343	
P3_SCM_R	0.2740	0.0016	0.0118	Interaction
P3_SCM_L	0.6197	<.0001	0.2011	
P3_UT_R	0.5542	0.0687	0.5147	
P3_UT_L	0.5181	0.0552	0.1118	
P4_LS_R	0.2859	0.0021	0.3770	
P4_LS_L	0.0136	0.0670	0.1278	Helmet1 different from 3
P4_ES_R	0.1919	<.0001	<.0001	Interaction
P4_ES_L	0.0351	<.0001	<.0001	Interaction
P4_SC_R	0.2108	0.0934	0.1952	
P4_SC_L	0.0020	0.0013	0.0616	Posture4 different from all except 6 and 7 (<0.05)
P4_SCM_R	0.0944	0.0024	0.1952	Posture3 different from all except 7 (<0.05)
P4_SCM_L	0.3522	<.0001	0.0305	Interaction
P4_UT_R	0.7960	0.3277	0.3070	
P4_UT_L	0.7103	0.2681	0.1662	
P5_LS_R	0.1786	0.2926	0.1250	
P5_LS_L	0.3255	0.3588	0.0382	Interaction
P5_ES_R	0.1756	0.0901	0.0037	
P5_ES_L	0.3024	0.0126	0.2216	

P5_SC_R	0.1290	0.3426	0.0082	Interaction
P5_SC_L	0.0979	0.3601	0.0012	Interaction
P5_SCM_R	0.4154	0.1426	0.0892	
P5_SCM_L	0.1228	0.6127	0.0216	Interaction
P5_UT_R	0.7049	0.2285	0.1945	
P5_UT_L	0.3147	0.2762	0.1280	

EMG – Median

	Helmet	Posture	Helmet*Posture	Pairwise Comparison
P1_LS_R	0.3736	0.4959	0.0038	Interaction
P1_LS_L	0.4976	0.3143	0.1924	
P1_ES_R	0.4723	0.6409	0.5761	
P1_ES_L	0.9527	0.5195	0.5696	
P1_SC_R	0.5696	0.6051	0.6952	
P1_SC_L	0.9707	0.5113	0.3542	
P1_SCM_R	0.6948	0.1881	0.0935	
P1_SCM_L	0.8504	0.0976	0.3495	
P1_UT_R	0.2138	0.1394	0.8022	
P1_UT_L	0.9849	0.2401	0.9095	
P2_LS_R	0.2977	0.4159	0.7673	
P2_LS_L	0.0812	0.0123	<.0001	Interaction
P2_ES_R	0.5220	0.0048	0.1519	Posture6 different from 3 and 4 (<0.05)
P2_ES_L	0.5462	0.0015	0.2717	Posture6 different from 3 and 4 (<0.05) Posture 7 different from 3 (<0.05)
P2_SC_R	0.3497	0.5058	0.1388	
P2_SC_L	0.0074	<.0001	0.0141	Interaction
P2_SCM_R	0.5574	0.0850	0.0980	
P2_SCM_L	0.5788	<.0001	0.2633	Posture2,3,4 different from 5,6,7 (<0.05)
P2_UT_R	0.7194	0.4412	0.3857	
P2_UT_L	0.6765	0.0963	0.1191	
P3_LS_R	0.1515	0.0041	0.1196	Posture 3 different from 4 and 5 (<0.05)
P3_LS_L	0.1935	0.0043	0.0104	Interaction
P3_ES_R	0.4409	<.0001	0.0003	Interaction
P3_ES_L	0.4197	<.0001	<.0001	Interaction
P3_SC_R	0.1732	0.0181	0.7419	Posture1,4,5,7 NOT different
P3_SC_L	0.0672	<.0001	0.3569	Posture7 different from all (<0.05) Posture5 different from 1 (<0.05)
P3_SCM_R	0.2251	0.0013	0.0159	Interaction
P3_SCM_L	0.6204	<.0001	0.1707	Posture7 different from all (<0.05) Posture5 different from 1,2,4,7 (<0.05) Posture3 different from 1 (<0.05)
P3_UT_R	0.5404	0.0648	0.4950	
P3_UT_L	0.5179	0.0589	0.0848	
P4_LS_R	0.3873	0.0044	0.6284	Posture3 different from 4 and 5 (<0.05)
P4_LS_L	0.0281	0.0749	0.0338	Interaction
P4_ES_R	0.2323	<.0001	<.0001	Interaction
P4_ES_L	0.1059	<.0001	<.0001	Interaction
P4_SC_R	0.2560	0.0884	0.4147	
P4_SC_L	0.0019	0.002	0.0342	Interaction
P4_SCM_R	0.3702	0.0174	0.5844	
P4_SCM_L	0.3784	0.0017	0.0123	Interaction
P4_UT_R	0.7932	0.4057	0.3751	
P4_UT_L	0.6615	0.2706	0.1460	
P5_LS_R	0.1409	0.3807	0.1041	

P5_LS_L	0.2842	0.3139	0.0337	Interaction
P5_ES_R	0.1303	0.0832	0.0028	Interaction
P5_ES_L	0.2725	0.0087	0.1351	Posture1 NOT different from 3
P5_SC_R	0.1311	0.3111	0.0234	Interaction
P5_SC_L	0.1145	0.3828	0.0122	Interaction
P5_SCM_R	0.4068	0.1888	0.0650	
P5_SCM_L	0.1005	0.5395	0.0203	Interaction
P5_UT_R	0.7361	0.2462	0.2182	
P5_UT_L	0.7361	0.2462	0.2182	

EMG – Onset

	Helmet	Posture	Helmet*Posture	Pairwise Comparison
P3_LS_R	0.5031	0.1431	0.5363	
P3_LS_L	0.7883	0.7504	0.1933	
P3_ES_R	0.7535	0.2142	0.7373	
P3_ES_L	0.0823	0.2051	0.6083	
P3_SC_R	0.2501	0.7317	0.474	
P3_SC_L	0.9696	0.0044	0.6241	Posture5,6 different from 2,4,7
P3_SCM_R	0.9355	0.1135	0.7866	
P3_SCM_L	0.3611	0.0037	0.6028	Posture5 different from all
P3_UT_R	0.8712	0.9642	0.0073	
P3_UT_L	0.7515	0.6091	0.5745	
P4_LS_R	0.4163	0.5224	0.1927	
P4_LS_L	0.5117	0.5342	0.2174	
P4_ES_R	0.8087	0.9006	0.2342	
P4_ES_L	0.917	0.8174	0.9597	
P4_SC_R	0.5295	0.2498	0.8479	
P4_SC_L	0.8305	0.2181	0.0829	
P4_SCM_R	0.1906	0.3206	0.7158	
P4_SCM_L	0.4043	0.0507	0.5575	
P4_UT_R	0.1345	0.1355	0.2539	
P4_UT_L	0.2357	0.38	0.6701	
P5_LS_R	0.8122	0.9642	0.56	
P5_LS_L	0.3285	0.5736	0.1718	
P5_ES_R	0.9942	0.5546	0.2189	
P5_ES_L	0.4563	0.2369	0.5197	
P5_SC_R	0.4309	0.1428	0.9164	
P5_SC_L	0.7517	0.7729	0.5584	
P5_SCM_R	0.2681	0.9174	0.2639	
P5_SCM_L	0.6361	0.6848	0.3322	
P5_UT_R	0.5168	0.7626	0.1053	
P5_UT_L	0.4244	0.0291	0.7455	Posture2 different from all

EMG – Peak

	Helmet	Posture	Helmet*Posture	Pairwise Comparison
P1_LS_R	0.8585	0.3922	0.0179	Interaction
P1_LS_L	0.5798	0.5885	0.2820	
P1_ES_R	0.1606	0.5536	0.6414	
P1_ES_L	0.8243	0.3537	0.6865	
P1_SC_R	0.0541	0.8564	0.0111	Interaction
P1_SC_L	0.9127	0.7676	0.3892	
P1_SCM_R	0.4297	0.4360	0.0479	Interaction
P1_SCM_L	0.7012	0.7995	0.2131	
P1_UT_R	0.1575	0.1140	0.6954	
P1_UT_L	0.9820	0.1646	0.9470	
P2_LS_R	0.3224	0.0075	0.6918	Posture6 different from 4 and 5 (<0.05)
P2_LS_L	0.2111	0.0025	0.0082	Interaction
P2_ES_R	0.4156	<.0001	0.1937	Posture2 different from 3 and 4 (<0.05) Posture3 different from all except 4 and 5 (<0.05) Posture6 different from all except 2 and 7 (<0.05) Posture7 different from all (<0.01)
P2_ES_L	0.2425	0.0004	0.4538	Posture6 and 7 different from 3 and 4 (<0.05)
P2_SC_R	0.2476	0.1336	0.6443	
P2_SC_L	0.0243	<.0001	0.0208	Interaction
P2_SCM_R	0.5971	0.0762	0.2058	
P2_SCM_L	0.1989	<.0001	0.3944	Posture2,3,4 different from 5,6,7 (P<0.01)
P2_UT_R	0.7387	0.4274	0.5897	
P2_UT_L	0.7219	0.0225	0.0871	Posture7 different from 2 and 4 (<0.05)
P3_LS_R	0.1068	0.0126	0.0381	Interaction
P3_LS_L	0.1551	0.0125	0.0013	Interaction
P3_ES_R	0.2020	<.0001	0.0042	Interaction
P3_ES_L	0.5139	<.0001	0.018	Interaction
P3_SC_R	0.4479	0.101	0.5253	
P3_SC_L	0.0442	<.0001	0.2499	Posture7 different from all except 5 (<0.01)
P3_SCM_R	0.6566	0.0022	0.0056	Interaction
P3_SCM_L	0.5135	<.0001	0.1539	Posture5 different from all except 3 and 6 (<0.05) Posture7 different from all (<0.05)
P3_UT_R	0.7738	0.0960	0.7062	
P3_UT_L	0.7129	0.0604	0.7897	
P4_LS_R	0.3077	0.003	0.1399	Posture3 different from 4 and 5 (<0.05)
P4_LS_L	0.009	0.1287	0.2833	
P4_ES_R	0.0228	<.0001	<.0001	Interaction
P4_ES_L	0.1669	<.0001	0.0556	Posture2 different from all except 6 (<0.05) Posture6 different from all except 2 (<0.05) Posture3,4,5 different from 2 and 6 (<0.05)
P4_SC_R	0.4321	0.0930	0.4485	
P4_SC_L	0.0261	0.0013	0.2004	Posture4 different from 2,3,5 (<0.05) Posture7 different from 1 (<0.05)
P4_SCM_R	0.0953	<.0001	0.0623	Posture2 different from all (<0.05)
P4_SCM_L	0.2931	<.0001	0.0729	Posture7 different from all (<0.01) Posture1 different from 2 (<0.05)

P4_UT_R	0.8779	0.1339	0.4958	
P4_UT_L	0.8168	0.3472	0.3760	
P5_LS_R	0.2031	0.5155	0.2820	
P5_LS_L	0.4872	0.5937	0.1289	
P5_ES_R	0.2980	0.1655	0.0967	
P5_ES_L	0.6667	0.0618	0.4585	
P5_SC_R	0.1254	0.7604	0.0874	
P5_SC_L	0.2582	0.4403	0.0054	Interaction
P5_SCM_R	0.5325	0.4138	0.4460	
P5_SCM_L	0.4326	0.8556	0.1929	
P5_UT_R	0.5065	0.2420	0.3097	
P5_UT_L	0.5817	0.3694	0.2123	

EMG – RMS

	Helmet	Posture	Helmet*Posture	Pairwise Comparison
P1_LS_R	0.5579	0.4859	0.0116	
P1_LS_L	0.5009	0.3637	0.2382	
P1_ES_R	0.3269	0.6272	0.5728	
P1_ES_L	0.9167	0.4433	0.6000	
P1_SC_R	0.1308	0.6204	0.6023	
P1_SC_L	0.9545	0.5725	0.3556	
P1_SCM_R	0.6467	0.2244	0.1056	
P1_SCM_L	0.8344	0.2118	0.3578	
P1_UT_R	0.2003	0.1313	0.7665	
P1_UT_L	0.9838	0.2231	0.9154	
P2_LS_R	0.2746	0.1809	0.7620	
P2_LS_L	0.1183	0.0116	0.0001	
P2_ES_R	0.5078	0.0019	0.099	Posture6 different from 3,4 (<0.01)
P2_ES_L	0.3087	0.001	0.2162	Posture6 different from 3,4 (<0.05) Posture7 different from 3 (<0.05)
P2_SC_R	0.4518	0.5224	0.4622	
P2_SC_L	0.0068	<.0001	0.0154	
P2_SCM_R	0.6862	0.0718	0.0398	
P2_SCM_L	0.3863	<.0001	0.4151	Posture1,2,3 different from 4,5,6 (<0.01)
P2_UT_R	0.734	0.5496	0.4246	
P2_UT_L	0.7047	0.0816	0.1084	
P3_LS_R	0.1361	0.0049	0.0664	Posture3 different from 4,5 (<0.05)
P3_LS_L	0.184	0.0052	0.0054	
P3_ES_R	0.4186	<.0001	0.0004	
P3_ES_L	0.421	<.0001	<.0001	
P3_SC_R	0.1866	0.0267	0.7331	Posture1,3,4,6 NOT different from each other
P3_SC_L	0.0719	<.0001	0.3124	Posture7 different from all (<0.05) Posture5 different from 1 and 7 (<0.05)
P3_SCM_R	0.2881	0.0017	0.0139	
P3_SCM_L	0.6016	<.0001	0.1918	Posture7 different from all (<0.01) Posture5 different from 1,2,4,7 (<0.01) Posture2,4 different from 5 (<0.01) Posture3 different from 1 (0.04)
P3_UT_R	0.5808	0.067	0.5332	
P3_UT_L	0.5231	0.0558	0.1387	
P4_LS_R	0.2699	0.0024	0.2641	Posture3 different from 4,5 (<0.05)
P4_LS_L	0.007	0.0858	0.1306	Helmet1 different from 3,4 (<0.05)
P4_ES_R	0.1457	<.0001	<.0001	
P4_ES_L	0.0311	<.0001	<.0001	
P4_SC_R	0.2141	0.1855	0.3385	
P4_SC_L	0.0035	0.0015	0.1024	Helmet1 different from 3,6 (<0.05)
P4_SCM_R	0.0773	0.0013	0.1339	Posture3 different from all except 7 (<0.01)
P4_SCM_L	0.3835	<.0001	0.0402	
P4_UT_R	0.8209	0.2352	0.2821	
P4_UT_L	0.7476	0.2692	0.1977	

P5_LS_R	0.1918	0.2629	0.1464	
P5_LS_L	0.3593	0.3661	0.045	
P5_ES_R	0.181	0.0892	0.0056	
P5_ES_L	0.3169	0.0126	0.2519	
P5_SC_R	0.1301	0.4002	0.0079	
P5_SC_L	0.1153	0.4026	0.0011	
P5_SCM_R	0.425	0.1596	0.105	
P5_SCM_L	0.1524	0.6592	0.0224	
P5_UT_R	0.6573	0.2178	0.1935	
P5_UT_L	0.3379	0.2857	0.1325	

EMG – Time to Max

	Helmet	Posture	Helmet*Posture	Pairwise Comparison
P1_LS_R	0.7308	0.7514	0.0035	
P1_LS_L	0.0386	0.3542	<.0001	
P1_ES_R	0.7982	0.2696	0.0665	
P1_ES_L	0.6633	0.6523	0.1689	
P1_SC_R	0.2422	0.77	0.1275	
P1_SC_L	0.0043	0.5605	<.0001	
P1_SCM_R	0.5588	0.7257	0.0154	
P1_SCM_L	0.0674	0.5039	<.0001	
P1_UT_R	0.3702	0.8427	0.0005	
P1_UT_L	0.1684	0.3284	0.0371	
P2_LS_R	0.0084	<.0001	0.4523	Helmet4 different from 6 (<0.01)
P2_LS_L	0.6085	<.0001	0.002	
P2_ES_R	0.0803	<.0001	0.4806	Posture6 different from all (<0.01) Posture7 different from all except 2 and 5 (<0.01) Posture2 different from 3,4,6 (<0.05) Posture5 different from 6 (<0.001)
P2_ES_L	0.0923	<.0001	0.0008	
P2_SC_R	0.6103	<.0001	0.997	Posture6,7 different from all but not from each other (<0.05)
P2_SC_L	0.4512	<.0001	0.8177	Posture7 different from all (<0.001) Posture4 different from 6 (<0.01)
P2_SCM_R	0.6824	<.0001	0.3607	Posture6,7 different from all but not from each other (<0.05)
P2_SCM_L	0.6824	<.0001	0.3607	Posture7 different from all (<0.001) Posture 5 different from 6 (<0.001)
P2_UT_R	0.7103	<.0001	0.178	Posture6 different from all except 7 (<0.05) Posture7 different from all except 6 and 2 (<0.01)
P2_UT_L	0.7016	<.0001	0.347	Posture6,7 different from all but not from each other (<0.05)
P3_LS_R	0.8812	<.0001	0.6565	Posture1 different from all except 3 and 7 (<0.05) Posture3 different from 5 and 6 (<0.01) Posture4 different from 1 and 6 (<0.05)
P3_LS_L	0.9806	<.0001	0.6941	Posture1 different from all (<0.01)
P3_ES_R	0.7695	<.0001	0.1592	Posture 1 different from all except 2 (<0.01) Posture2 different from all except 1 and 6 (<0.05)
P3_ES_L	0.8357	<.0001	0.4802	Posture1 different from 3,4,7 (<0.01) Posture3,7 different from 1,2,6 (<0.05) Posture4 different from 1 (<0.01) Posture2,6 different from 3,7 (<0.01)
P3_SC_R	0.0549	0.0021	0.8251	Posture1 different from 5,6,7 (<0.05)
P3_SC_L	0.1014	0.0008	0.86	Posture1 different from 5,6,7 (<0.05)
P3_SCM_R	0.9161	<.0001	0.9747	Posture1 different from all except 3 (<0.01) Posture2,7 different from 1 (<0.01) Posture3 different from 5,6 (<0.01) Posture5,6 different from 1,3 (<0.01)
P3_SCM_L	0.8814	<.0001	0.5899	Posture1 different from 2,5,6 (<0.05) Posture3,4 different from 5,6 (<0.01) Posture6 different from all except 5 (<0.01) Posture7 different from 6 (<0.01)

P3_UT_R	0.956	<.0001	0.613	Posture1 different from all except 4 (<0.01) Posture4 different from 6,7 (<0.05)
P3_UT_L	0.6154	0.0028	0.0796	Posture1 different from 2,6,7 (<0.05)
P4_LS_R	0.014	0.0199	0.4916	Helmet1 different from 5 Posture3 different from 2 and 6 (<0.05)
P4_LS_L	0.3309	<.0001	0.4714	Posture6 different from all except 2 (<0.05) Posture7 different from 6 (<0.001)
P4_ES_R	0.0867	<.0001	0.3542	Posture3 different from all except 7 (<0.05) Posture7 different from 4 and 5 (<0.05)
P4_ES_L	0.3868	<.0001	0.4786	Posture3 different from all (<0.05) Posture 4,5,6,7 different from all except 2 (<0.05)
P4_SC_R	0.9004	0.0477	0.0523	Posture2 different from 3
P4_SC_L	0.8331	<.0001	0.7731	Posture2 different from all except 6 (<0.05)
P4_SCM_R	0.1746	0.001	0.0359	
P4_SCM_L	0.662	<.0001	0.18	Posture2 different from all (<0.05) Posture3 different from 2,5 (<0.05) Posture4 different from 2,5,7 (<0.05) Posture6 different from 2 (<0.001)
P4_UT_R	0.3522	0.2009	0.214	
P4_UT_L	0.5103	0.0011	0.0875	Posture6 different from 5,7 (<0.01)
P5_LS_R	0.252	0.0856	0.748	
P5_LS_L	0.8307	0.1188	0.7604	
P5_ES_R	0.6608	0.0456	0.8685	
P5_ES_L	0.433	0.0405	0.6431	
P5_SC_R	0.3088	0.2269	0.9321	
P5_SC_L	0.8073	0.0941	0.9381	
P5_SCM_R	0.021	0.0272	0.7298	
P5_SCM_L	0.8961	0.0931	0.8636	
P5_UT_R	0.6574	0.0419	0.937	
P5_UT_L	0.579	0.0247	0.865	

EMG – MCI

	Helmet	Posture	Helmet*Posture	Pairwise Comparison
P1	0.4512	0.2108	0.8672	
P2	0.7012	<.0001	0.4459	
P3	0.3285	<.0001	0.6171	Posture7 different from all (<0.05)
P4	0.0922	0.0246	0.3683	Posture2,3,4,5 not significantly different (<0.05) Posture6,7 different
P5	0.2987	0.2797	0.0212	Interaction